

## Verkkosivut MCV Remonttipalvelulle

Emma Eklund  
Mari Khamaiseh



<b>Tekijä(t)</b> Emma Eklund, Mari Khamaiseh	
<b>Koulutusohjelma</b> Tietojenkäsittelyn koulutusohjelma	
<b>Raportin/Opinnäytetyön nimi</b> Verkkosivut MCV Remonttipalvelulle	<b>Sivu- ja liitesivumäärä</b> 70 + 23
<p>Työn tavoitteena on suunnitella ja toteuttaa julkaisuvalmis verkkosivusto remontointialan pienyritykselle käyttäen ohjelmointitekniikoita HTML, CSS, JavaScript, Bootstrap, Lightbox ja PHP. Työ rajataan käsittelemään ainoastaan sivuston suunnittelua ja toteuttamista, eikä julkaisuun liittyviä asioita käsitellä ollenkaan. Työ toteutetaan noin kuuden kuukauden aikana syksyn 2017 ja kevään 2018 aikana.</p> <p>Verkkosivuja luodaan joko alusta lähtien ohjelmoimalla muun muassa HTML- ja CSS-tekniikoita käyttäen tai luomalla sivusto sisällönhallintajärjestelmällä, kuten WordPressillä.</p> <p>Verkkosivut ovat tärkeä osa yrityksen Internet-markkinointia, sillä nykypäivänä ihmiset oletavat löytävänsä yrityksen verkosta. Yrityksen verkkosivustoa tulee myös mainostaa ja tehokkain tapa ihmisten tavoittamiseen on hakukoneoptimointi, jonka tarkoitus on nostaa yrityksen sivusto paremmalle sijoitukselle hakukoneiden, kuten Googlen, tuloksissa.</p> <p>Ennen sivuston suunnittelun aloitusta on hyvä tehdä vaatimusmäärittely, jotta saadaan selkeä kuva siitä, mitä sivustolta halutaan, mikä on lähtötilanne ja mitä sivustolla pitäisi pystyä tekemään. Sen lisäksi yrityksen on tärkeää kartoittaa ja määritellä oma(t) kohderyhmänsä, minkä tavoitteena on löytää yrityksen tärkeimmät asiakasryhmät, jotta yritys pystyy erottumaan kilpailijoistaan ja houkuttelemaan ihmiset oman yrityksensä asiakkaisiksi.</p> <p>Verkkosivustoa suunniteltaessa on hyvä tutustua myös siihen, millainen on hyvä käyttäjäkokemus ja mitä sen eteen voi tehdä. Täydellisen käyttäjäkokemuksen tuottaa tuote tai palvelu, joka täyttää käyttäjän kaikki tarpeet helposti.</p> <p>Ennen sivuston suunnittelun aloittamista on hyvä tehdä myös vertailuanalyysia eli benchmarkingia, jonka avulla yrityksen toimintaa ja prosesseja vertaillaan muihin yrityksiin ja pyritään löytämään niistä parhaita käytäntöjä ja vahvoja puolia, joita voidaan hyödyntää omassa liiketoiminnassa.</p> <p>Tämä opinnäytetyö on toiminnallinen työ ja sen tuotos on julkaisuvalmis sivusto. Sivuston suunnittelu aloitetaan luomalla PowerPoint-ohjelmalla sivustosta prototyyppi, jolle tehdään think aloud -käytettävyydesti. Sen jälkeen prototyypistä tehdään visuaalinen suunnitelma PhotoShop-ohjelmalla ja näitä kahta mallina käyttäen ohjelmoidaan sivusto. Lopulta myös valmiille sivustolle tehdään think aloud -käytettävyydesti ja testin tulosten mukaan vaadittavat korjaukset. Opinnäytetyö sisältää käyttöohjeet, jotta toimeksiantaja voi tehdä itse pieniä päivityksiä sivustolle.</p>	
<b>Asiasanat</b> WWW-sivut, HTML, CSS, ohjelmointi, käyttöohjeet	

# Sisällys

1	Johdanto .....	1
2	Verkkosivujen suunnittelu.....	3
2.1	Verkkosivujen suunnittelun historiaa .....	3
2.2	Käytettävät ohjelmat ja työkalut.....	4
3	Teknologioita.....	6
3.1	HTML.....	6
3.2	CSS .....	6
3.3	JavaScript .....	7
3.4	Bootstrap .....	7
3.5	Lightbox .....	8
3.6	PHP .....	8
4	Sisällönhallintajärjestelmät .....	9
4.1	Toimintaperiaate .....	9
4.2	Yleisimmät sisällönhallintajärjestelmät .....	9
5	Verkkosivujen käyttäminen markkinoinnissa .....	11
6	Vaatimusmäärittely.....	13
7	Käyttäjäkokemus.....	14
8	Benchmarking .....	16
9	Kohderyhmän kartoitus ja määrittely .....	19
10	Käyttöohjeiden tuottaminen .....	21
11	MCV Remonttipalvelun verkkosivusto .....	23
11.1	Yrityksen esittely .....	23
11.2	Projektisuunnitelma.....	23
11.3	Sivuston toteutustavan valintaperuste .....	24
11.4	Käyttöohjeistus asiakkaalle .....	24
11.5	Toteutus.....	24
11.5.1	Prototyyppi .....	25
11.5.2	Think aloud -käytettävyydesti .....	30
11.5.3	Sivuston visuaalinen suunnitelma .....	32
11.5.4	Sivuston ohjelmointi .....	33
11.6	Tuotos.....	54
12	Pohdinta.....	63
	Lähteet .....	65
	Liitteet.....	71
	Liite 1. Keskeiset käsitteet .....	71
	Liite 2. Projektisuunnitelma.....	73
	Liite 3. Käyttöohjeet: GIMP – Asennus .....	74

Liite 4. Käyttöohjeet: GIMP – Kuvankäsittely .....	77
Liite 5. Käyttöohjeet: Notepad++ – Asennus.....	82
Liite 6. Käyttöohjeet: Notepad++ – Tiedoston käsittely .....	88
Liite 7. Käyttöohjeet: Kuvagalleria .....	90
Liite 8. Sivuston visuaalinen suunnitelma .....	93

# 1 Johdanto

Työn tavoitteena on suunnitella ja toteuttaa julkaisuvalmis verkkosivusto pienyritykselle. Sivusto toteutetaan alusta lähtien käsin ohjelmoimalla käyttäen ohjelmointitekniikoita HTML, CSS, JavaScript, Bootstrap, Lightbox ja PHP. HTML, CSS ja JavaScript muodostavat verkkosivuston pohjan, jossa HTML-merkkintäkielellä luodaan sivuston perusrakenne, CSS:llä sivuston ulkoasu ja JavaScriptillä sivuston toiminnallisuus. Sivustoa täydennetään Bootstrapilla, jolla luodaan kuvakaruselli, ja jota hyödynnetään myös lomakkeen luomisessa. Kuvagallerian luomisessa hyödynnetään Lightbox-tekniikkaa ja lomakkeen tietojen keräämisessä PHP-ohjelmointikieltä.

Opinnäytetyön teoriaosuudessa käydään läpi verkkosivujen suunnittelun historiaa sekä käytettäviä ohjelmointiteknologioita ja sisällönhallintajärjestelmiä sekä etsitään vastauksia seuraaviin kysymyksiin:

- Miten verkkosivuja voi hyödyntää markkinoinnissa?
- Mitä on vaatimusmäärittely?
- Millainen on hyvä käyttäjäkokemus?
- Mitä on benchmarking?
- Mitä on segmentointi?
- Millainen on hyvä käyttöohje?

Verkkosivuja voi luoda joko alusta lähtien ohjelmoimalla, muun muassa HTML- ja CSS-tekniikoita käyttäen, tai luomalla sivusto sisällönhallintajärjestelmällä, kuten WordPressillä. Jos koodin kirjoittaa alusta lähtien itse, tarvitsee jonkinlaisen tekstieditorin, jolla voi kirjoittaa HTML-koodia, kuten Eclipse tai Notepad++. On olemassa myös ohjelmia, jotka luovat koodin itse, tällaisia ohjelmia kutsutaan nimellä WYSIWYG (What You See Is What You Get) ja niitä ovat muun muassa Adobe Dreamweaver, joka on erillinen ohjelmisto, sekä Wix ja Weebly, jotka ovat verkkoselaimen kautta toimivia ohjelmia. Sisällönhallintajärjestelmiä löytyy sekä ohjelmointia osaavalle että osaamattomalle, muun muassa WordPress ja Drupal.

Nykypäivänä asiakkaat olettavat löytävänsä yrityksen myös verkosta, joten verkkosivut ovat tärkeä osa yrityksen Internet-markkinointia. Sivujen sisältö tulee suunnitella yrityksen kohderyhmän mukaan ja ne vaativat myös mainontaa, jotta mahdolliset asiakkaat löytäisivät sivut ja yrityksen. Tehokkain tapa ihmisten tavoittamiseen on hakukoneoptimointi, jonka tarkoituksena on nostaa yritys paremmalle sijoiutukselle hakukoneiden tuloksissa. Sitä varten verkkosivuston sisältöön upotetaan sivustoa kuvaavia hakusanoja ja -lauseita.

Ennen sivuston suunnittelun aloitusta on hyvä tehdä vaatimusmäärittely, jotta saadaan selkeä kuva siitä, mitä sivustolta halutaan, mikä on lähtötilanne ja mitä sivustolla pitäisi

pystyä tekemään. Sen lisäksi yrityksen on tärkeää kartoittaa ja määritellä oma(t) kohde-ryhmänsä, mitä kutsutaan segmentoinniksi. Segmentoinnin tavoitteena on löytää yrityksen tärkeimmät asiakasryhmät, jotta yritys pystyisi erottumaan kilpailijoista ja varmistamaan sen, että asiakkaat valitsevat oman yrityksen palvelut. Segmentointi on tärkeää myös verkkosivujen suunnittelussa, jotta sisällön ja yleisilmeen kehittäminen lähtee heti oikeaan suuntaan. Segmentointia voi käyttää sekä nykyisiin että mahdollisiin tuleviin asiakkaisiin ja perusteina voi käyttää esimerkiksi ikää, toimipaikkaa, harrastuksia ja mieltymyksiä.

Verkkosivustoa suunniteltaessa on hyvä tutustua myös siihen, millainen on hyvä käyttäjäkokemus ja mitä sen eteen voi tehdä. Peter Morvillen mukaan käyttäjäkokemus rakentuu seitsemästä elementistä: sivuston arvosta käyttäjälleen, sen hyödyllisyydestä, käytettävyydestä, haluttavuudesta, tavoitettavuudesta, esteettömyydestä sekä uskottavuudesta. Kaikkien seitsemän osion on toimittava, jotta käyttäjäkokemuksesta voisi tulla hyvä. Käyttäjäkokemus muodostuu käyttäjän huomioista, toiminnasta sekä ymmärryksestä, ja tunne-reaktiot prosessin aikana määrittävät, tuleeko kokemuksesta positiivinen vai negatiivinen. Ennen suunnittelun aloittamista on hyvä myös tehdä vertailuanalyysia eli benchmarkingia muiden yritysten kanssa. Benchmarking-menetelmän avulla oman yrityksen toimintaa ja prosesseja vertaillaan muihin yrityksiin ja pyritään löytämään niistä parhaita käytäntöjä ja vahvoja puolia, joita voi hyödyntää omassa liiketoiminnassa. Benchmarkingia voi toteuttaa esimerkiksi tutustumalla vertailun kohteena olevaan yritykseen paikan päällä tai etsimällä siitä tietoa verkosta, kirjoista tai lehdistä. Koska asiakkaan on tarkoitus pystyä itse tekemään pieniä päivityksiä sivustoon, käydään teoriaosuudessa myös läpi, millainen on hyvä ohje.

Sivuston varsinainen suunnittelu aloitetaan luomalla tulevasta sivustosta prototyyppi, jolle tehdään think aloud -käytettävyydesti. Testin jälkeen prototyypistä tehdään visuaalinen suunnitelma ja näiden kahden tuotoksen avulla aloitetaan ohjelmoimaan sivustoa. Think aloud -käytettävyydesti toistetaan vielä lopuksi valmiille sivustolle ja tehdään sivustoon tarvittavat korjaukset ennen sivuston luovuttamista asiakkaalle. Työn empiiriseen osaan tehdään myös kirjalliset ohjeet asiakkaalle pieniä päivityksiä varten. Opinnäytetyö on toiminnallinen työ ja sen tuotos on julkaisuvalmis sivusto. Työ rajataan käsittelemään ainoastaan sivuston suunnittelua ja toteuttamista, eikä julkaisuun liittyviä asioita käsitellä ollenkaan. Työn keskeiset käsitteet on selitetty liitteessä 1. Työn toimeksiantaja on Vantaalla sijaitseva pienyritys MCV Remonttipalvelu ja Kotiapu.

## 2 Verkkosivujen suunnittelu

Verkkosivustojen kehittämistä voi kuvata kolmella eri tasolla: rakenne, tyyli ja käyttäytyminen. Rakenne- eli sisältötaso tarkoittaa sivuston HTML-koodia, jonka päälle rakennetaan sivuston muut osat. HTML-rakenne voi koostua tekstistä, kuvista ja multimedialta ja se sisältää myös sivuston navigoinnin eli linkit sivulta toiselle. Tyyli- eli esitystaso kertoo, miltä rakennettu HTML-dokumentti näyttää vierailijan silmissä. Tämä taso määritetään CSS-tekniikalla ja siinä voidaan ottaa huomioon myös sivuston ulkoasu eri näyttöko'illa ja laitteissa. Käyttäytymistasolla verkkosivu reagoi käyttäjän tekemiin toimintoihin tai tekee sivuun muutoksia asetettujen ehtojen mukaan. Tämä taso toteutetaan usein JavaScriptillä. Verkkosivuston kehittämisen kolme tasoa on esitetty kuvassa 1. (Kyrnin 2017a.)



Kuva 1. Verkkosivuston kehittämisen kolme tasoa (Kyrnin 2017a)

### 2.1 Verkkosivujen suunnittelun historiaa

Ensimmäiset verkkosivustot olivat täysin tekstipohjaisia ilman merkittävää asemointia lukuun ottamatta otsikoita ja kappaleita. Elokuussa 1991 brittiläinen tietojenkäsittelytieteen tutkija Tim Berners-Lee julkaisi ensimmäisen verkkosivuston, joka oli HTML:llä tehty yksinkertainen tekstipohjainen sivu, jolla oli muutama linkki. HTML:n ensimmäiset versiot mahdollistivat ainoastaan perusrakenteen luomisen: otsikot, kappaleet ja linkit. (Chapman 2009.) Alussa verkkosivujen suunnittelu oli hyvin minimalistista, mutta vähitellen HTML:stä tuli monimutkaisempaa ja joustavampaa ja käyttäjät pystyivät lisäämään kuvia ja tauluja verkkosivuille (Astra Design Company).

Vuonna 1993 julkaistiin ensimmäinen verkkoselain, Mosaic. Mosaicilla pystyi selaamaan Internetiä graafisesti ja se toi Internetin kaikkien ulottuville. Selaimen avulla pystyi katselemaan tekstiä ja grafiikkaa, mutta sivuston asettelu oli vielä hyvin rajoittunutta. (Astra Design Company.) Vuonna 1994 perustettu World Wide Web Consortium (W3C) teki HTML:stä standardin verkkosivujen merkkauksiensa (Chapman 2009). HTML ei kuitenkaan taipunut kaikkeen, joten sen ohelle kehitettiin vuonna 1995 JavaScript. Vuonna 1996 kehitettiin Flash, jolla pystyi luomaan erilaisia muotoja, asetteluja ja animaatioita sekä

käyttämään erilaisia fontteja. Vuonna 1998 syntyi parempi tekniikka rakenteen suunnitteluun – CSS – joka mahdollisti sisällön ja ulkoasun erottamisen toisistaan. (Ruluks 2014.) CSS:n syntyminen toi nopeaa kehitystä verkkosivujen suunnitteluun (Astra Design Company).

2000-luvun ensimmäisellä kymmenyksellä älypuhelimien yleistymisen toi mukanaan tarpeen kehittää tekniikoita responsiiviseen suunnitteluun eli sivuston skaalautumiseen pienelle näytölle sopivaksi. Tätä tarkoitusta varten keksittiin muun muassa jakaa sivu sarakkeisiin. (Ruluks 2014.)

## **2.2 Käytettävät ohjelmat ja työkalut**

Verkkosivujen suunnitteluun löytyy paljon oppimateriaalia sekä kirjoissa että Internetissä. Suurin osa materiaalista on englanniksi, mutta myös suomenkielistä kirjallisuutta on saatavilla. Yksi kattavimpia Internet-sivustoja verkkosivujen suunnittelun opiskeluun on W3schools-sivusto osoitteessa <https://www.w3schools.com/>. Sivusto on englanninkielinen ja se käy kattavasti läpi useita eri teknologioita, kuten HTML, CSS ja JavaScript sekä serveripuolelta SQL, PHP ja ASP. (W3schools.com 1999-2018a.) Sivustolla voi myös suorittaa sertifikaatteja eri teknologioista. Opiskelu on maksutonta ja sertifikaatin saa osallistumalla kokeeseen verkossa ja maksamalla 95 dollaria (noin 80 euroa). (W3schools.com 1999-2018b.)

Verkkosivuja voi tehdä kahdella eri tavalla: joko käsin ohjelmoimalla – eli kirjoittamalla itse kaikki HTML-koodi, JavaScript ja muut – tai käyttämällä ohjelmaa, joka luo koodin käyttäjän puolesta. Käsin ohjelmointiin tarvitaan jonkinlainen tekstieditori, jolla voi kirjoittaa HTML-koodia. HTML-editoreja ovat esimerkiksi Eclipse ja Notepad++ (Kyrnin 2018). Koodia voi tuottaa kuitenkin lähes millä tahansa tekstinkäsittelyohjelmalla. Ohjelmia, jotka luovat koodin itse, kutsutaan myös nimellä WYSIWYG (What You See Is What You Get). WYSIWYG-editoreja ovat esimerkiksi Adobe Dreamweaver ja Microsoft Expression Studio Web Pro, jotka ovat erillisiä ohjelmistoja (Kyrnin 2017b). WYSIWYG-editoreiksi kutsutaan myös Internetissä toimivia ohjelmia, joilla saa kuka tahansa tehtyä kotisivut ilman mitään tuntemusta HTML-ohjelmoinnista. Nämä ohjelmat ovat usein ilmaisia, mutta tarjoavat maksullisia lisäpalveluita, kuten oman domain-nimen (omanimi.fi) sekä edistyneempää hakukoneoptimointia ja ammattimaisia sähköpostimarkkinointipalveluja. Tällaisia ohjelmia ovat esimerkiksi Wix, Weebly sekä Shopify. (Ivanovs 2017.) Ohjelmoinnin eri teknologioista kerrotaan tarkemmin luvussa 3.



Verkkosivustoja voi luoda myös erilaisilla sisällönhallintajärjestelmillä (Content Management System, CMS), joita on tarjolla monelle eri osaamistasolle. HTML- ja CSS-taitoja osaavalle löytyy esimerkiksi Drupal ja koodia ei-hallitsevalle Webydo. WordPress taas tarjoaa vaihtoehdon siltä väliltä, eli koodia ei tarvitse ymmärtää ollenkaan, mutta jos ymmärtää, voi omaa sivustoa muokata myös ohjelmoimalla. (Girard 2017.) Eri sisällönhallintajärjestelmistä kerrotaan luvussa 4.

Verkkosivujen suunnittelussa tarvitaan myös tuntemusta esimerkiksi kuvankäsittelyohjelmista, joita ovat esimerkiksi maksulliset Adobe Photoshop ja Adobe Illustrator sekä ilmaisohjelma GIMP.

### 3 Teknologioita

Verkkosivustojen kehittämisessä käytetään erilaisia teknologioita, joista tässä luvussa esitellään HTML, CSS, JavaScript, Bootstrap, Lightbox sekä PHP.

#### 3.1 HTML

HTML (HyperText Markup Language) on merkintäkieli, jolla luodaan verkkosivuja ja kuvataan verkkosivujen rakennetta (W3schools.com 1999-2018c). HTML:n avulla luodaan sähköisiä dokumentteja eli sivuja, jotka yhdistetään toisiinsa hyperlinkkien avulla. HTML-koodin avulla Internetselain osaa asemoida tekstin ja kuvat juuri kuten sivujen suunnittelija on tarkoittanut. HTML luo sivulle myös perusrakenteen, jonka ulkonäköä CSS-teknologia muuntelee. HTML on ikään kuin verkkosivuston luusto ja CSS sen iho. (Computer Hope 2017a.)

Sivusto rakentuu tageista, jotka merkitään merkkien < ja > sisään. <html>-tagien sisään tulee <head>-tagit ja <body>-tagit. <head>-tagien sisään kirjataan tietoa sivusta, kuten sivuston nimi ja missä CSS-tyylitiedosto sijaitsee ja <body>-tagien sisään kaikki, mikä näkyy selaimessa (kuva 2). HTML-koodia voi muokata millä tahansa tekstieditorilla, kunhan sen tallentaa .htm- tai .html-tiedostopäätteellä. (Computer Hope 2017a.)

```
<!DOCTYPE html>
<html lang="fi">
  <head>
    <title>MCV</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="stylesheet" href="css/style.css" type="text/css">
  </head>
  <body>
    <h1>Tähän sivun otsikko</h1>
    <p>Tähän sivun teksti.</p>
  </body>
</html>
```

Kuva 2. Sivuston perusrakenne HTML-koodina

#### 3.2 CSS

CSS (Cascading Style Sheet) -teknologian loi vuonna 1994 norjalainen Håkon Wium Lie (Computer Hope 2017b). Se on teknologia, jolla saa muokattua sivuston ulkonäköä ja rakennetta. CSS:n avulla pystyy kontrolloimaan muun muassa sivuston värejä, fontteja, sarakkeiden kokoa ja asemointia sekä miltä sivusto näyttää eri laitteilla katsottuna. (Tutorials Point 2018.)

CSS-tyylitiedoston tiedostopääte on .css ja tiedosto pitää linkittää sivulle merkitsemällä se <head>-tagien sisään sivun koodissa (kuva 2). CSS-tyylitiedoston voi luoda millä tahansa tekstieditorilla. (Computer Hope 2017c.) CSS on helposti opittavissa ja ymmärrettävissä ja sen avulla saa hallittua tehokkaasti HTML-dokumentin ulkonäköä. CSS tarjoaa paljon laajemman valikoiman eri määritteitä, joilla sivuston ulkonäköä voi parantaa, kuin HTML, ja sivuston ulkoasun hallinnointi on nopeaa ja helppoa CSS:n avulla. (Tutorials Point 2018.)

### 3.3 JavaScript

JavaScript on ohjelmointikieli, jonka kehitti amerikkalainen Brendan Eich (Computer Hope 2017d). Se on HTML:n ja CSS:n ohella yksi niistä teknologioista, jotka jokaisen websuunnittelijan tulisi hallita. JavaScriptiä käytetään ohjelmoimaan, miten sivusto käyttäytyy. (W3schools.com 1999-2018d.)

JavaScript sisällytetään yleensä HTML- tai ASP-tiedostoon ja sitä käytetään erilaisten toimintojen toteuttamiseen, joita pelkällä HTML-koodilla ei pysty toteuttamaan, kuten päivämäärän näyttäminen tai kalenterin luominen. JavaScript-koodin voi tallentaa myös erilliseen tiedostoon, jonka tiedostopääte on yleensä .js. Jotta koodi toimii, täytyy selaimen pystyä lukemaan ja suorittamaan JavaScriptiä. (Computer Hope 2017e.)

### 3.4 Bootstrap

Bootstrap on verkkosovelluskehityksen työkalu, jonka loivat Mark Otto ja Jacob Thornton. Siitä on tullut yksi tärkeimmistä työkaluista websivujen suunnittelijoille. Bootstrap on valtava kokoelma HTML-, CSS- ja JavaScript-koodia, jonka avulla voi myös luoda nopeasti responsiivisia sivustoja. Bootstrapin etuja on esimääritelty ruudukkojärjestelmä, jolla sivuston saa nopeasti responsiiviseksi. Ruudukko koostuu 12 sarakkeesta, joiden avulla sivun voi jakaa erikokoisiin osiin, esimerkiksi puoliksi eli kahteen sarakkeeseen (6 + 6) tai neljään saman levyiseen sarakkeeseen (3 + 3 + 3 + 3) tai vaikka kolmeen osaan, joista kaksi on saman levyistä ja yksi kapeampi (5 + 5 + 2). Bootstrapin avulla on helppo tehdä myös kuvista responsiivisia, sillä työkaluun sisältyy koodi automaattiseen kuvien koon muuntamiseen riippuen näytön koosta. Bootstrap sisältää useita eri komponentteja, joita lisäämällä sivustolle saa näyttäviä ja responsiivisia elementtejä, kuten navigointipalkkeja ja alasvetovalikkoja. Bootstrapin mukana tulee myös yli 10 eri jQuery pluginia, joilla sivustolle saa lisättyä interaktiivisuutta, kuten siirtymiä ja kuvakaruselleja. Bootstrap on hyvin dokumentoitu ja jokainen koodinpätkä on selitetty tarkkaan Bootstrapin kotisivuilla <https://getbootstrap.com/>. Sivuilta löytyy myös koodiesimerkkejä, joilla aloittelijakin pääsee alkuun. (Ouellette 2017.)

Bootstrapin huonoja puolia on sen koko, mutta sitä voi hallita lataamalla omalle sivustolle vain ne ominaisuudet, joita tarvitsee. Bootstrapin vahvuuksia on myös laaja joukko suunnittelijoita ja kehittäjiä sen takana, jotka kehittävät koodia edelleen ja joilta saa apua ongelmiin Bootstrapin kanssa. Bootstrapiin on saatavilla myös valmiita pohjia, joiden avulla sivustojen kehittäminen on vielä nopeampaa. (Ouellette 2017.) Bootstrap on luotu nykyajan selaimille ja on yhteensopiva kaikkien tämän päivän selainten kanssa, kuten Mozilla Firefox, Google Chrome ja Internet Explorer. Bootstrap on avoimen lähdekoodin työkalu, joten sen lataaminen ja käyttö on täysin ilmaista. (Tutorial Republic 2018.)

### **3.5 Lightbox**

Lightbox on JavaScript-kirjasto, joka avaa kuvat ja videot näytölle himmentäen muun sivun taustalle. Lightboxin kehitti Lokesh Dhakar ja se on avoimen lähdekoodin kirjasto. Tekniikka on saanut suuren suosion sen yksinkertaisen, mutta tyylikkään ulkoasun takia. (Wikipedia 2017.)

Lightbox ladataan Dhakarın sivustolta <http://lokeshdhakar.com/projects/lightbox2/> ja se on yhteensopiva yleisimpien selainten kanssa, kuten Internet Explorer, Google Chrome ja Mozilla Firefox. Lightbox käyttää apunaan jQueryä, joten se tulee myös ladata sivulle. Lightbox-tekniikkaa voi käyttää vapaasti sekä kaupallisiin että ei-kaupallisiin tarkoituksiin. (Dhakar.)

### **3.6 PHP**

PHP (PHP: Hypertext Preprocessor) on laajasti käytössä oleva avoimen lähdekoodin palvelinpuolen ohjelmointikieli, jota voi vapaasti ladata ja käyttää. Se sopii hyvin aloittelevalle ohjelmoijalle, kunhan hallussa on ainakin perusteet HTML:stä, CSS:stä ja JavaScriptistä. PHP:tä käyttää esimerkiksi WordPress ja Facebook. PHP toimii useilla eri alustoilla, kuten Windows, Linux, Unix ja Mac OS X ja on yhteensopiva lähes kaikkien nykypäivän palvelinten kanssa, kuten Apache ja IIS. (W3schools.com 1999-2018e.)

PHP-tiedostot voivat sisältää tekstiä sekä HTML-, JavaScript- ja PHP-koodia ja niiden tiedostopääte on .php. PHP:llä voidaan luoda dynaamista sisältöä verkkosivuille ja se voi luoda, avata, lukea, kirjoittaa, poistaa ja sulkea tiedostoja palvelimella. PHP:tä käytetään myös lomakkeiden tietojen keräämiseen, evästeiden lähettämiseen ja vastaanottamiseen sekä tietokannan tietojen muuttamiseen. PHP:n avulla näytölle voi tulostaa HTML:n lisäksi kuvia, pdf-tiedostoja, Flash-elokuvia sekä XHTML- ja XML-tekstiä. (W3schools.com 1999-2018e.)

## 4 Sisällönhallintajärjestelmät

Sisällönhallintajärjestelmä (engl. Content Management System eli CMS) on verkkopohjainen sovellus, jolla luodaan ja hallitaan nettisivuston sisältöä. Yleisimmin käytetyt sisällönhallintajärjestelmät ovat WordPress, Joomla ja Drupal. Muita sisällönhallintajärjestelmiä ovat muun muassa Episerver, DNN, Hubspot, Squarespace, Magento, Blogger, Adobe Dreamweaver, Wix, FrontPage ja Weebly. (Kauppinen 2015; W3Techs 2009-2018a.) Kaikkiaan eri sisällönhallintajärjestelmiä on satoja, mutta WordPress on ehdottomasti suosituin.

### 4.1 Toimintaperiaate

Sisällönhallintajärjestelmän avulla verkkosivustoja pystyy luomaan, muokkaamaan ja ylläpitämään suoraan selaimen kautta ilman minkäänlaisia ohjelmointitaitoja (Omni Partners Oy 2018). Sisällönhallintajärjestelmät sisältävät kaksi eri komponenttia: CMA (Content Management Application) ja CDA (Content Delivery Application). CMA on graafinen käyttöliittymä, jolla käyttäjä pystyy luomaan, muokkaamaan ja poistamaan sisältöä verkkosivulta ilman ohjelmointitaitoja. CDA tarjoaa back-end-palvelun, joka tukee sisällön hallintaa sivujen luomisen jälkeen. (Rouse 2016.)

Sisällönhallintajärjestelmät sisältävät useita eri osioita, jotka vaihtelevat eri järjestelmissä. Perustoiminnot ovat indeksointi, haku, tallennusmuodon hallinta, päivitys ja julkaisu. Muita sisällönhallintajärjestelmien ominaisuuksia ovat muun muassa hakukoneystävälliset URLit, keskustelufoorumit sekä helpot asennustoiminnot. (Rouse 2016.)

Sisällönhallintajärjestelmän valinnassa tulee ottaa huomioon monia asioita. Tärkeimpiä asioita ovat helppokäyttöinen käyttöliittymä (editor interface) ja hyvät hakutoiminnot. Muita asioita, joita valinnassa on hyvä miettiä, on esimerkiksi, kuinka monta ihmistä tulee käyttämään sovellusta, tarvitaanko monikielistä tukea sekä kuinka paljon hallintaoikeuksia pääkäyttäjille ja loppukäyttäjille annetaan. (Rouse 2016.)

### 4.2 Yleisimmät sisällönhallintajärjestelmät

Suosituimmat ja yleisimmät sisällönhallintajärjestelmät ovat Joomla, Drupal ja WordPress ja ne ovat kaikki open source eli avoimen lähdekoodin järjestelmiä ja täten ilmaisia järjestelmiä, joita kuka tahansa voi käyttää (Tuominen 2013).

Joomla on ollut käytössä vuodesta 2005 asti, mutta sen suosio on ollut viime vuosina laskusuunnassa ja moni Joomlaan käyttäjä on siirtynyt WordPressiin. Joomlaan kehityksessä

käytettäviä tekniikoita ovat PHP ja MySQL-tietokanta ja se sopii pieniin ja keskisuuriin projekteihin. (Kauppinen 2015.)

WordPress on maailman käytetyin sisällönhallintajärjestelmä ja sitä käyttää lähes kolmannes kaikista nettisivuista (W3Techs 2009-2018a). WordPress on ollut nopeimmin kasvava sisällönhallintajärjestelmä jo kahdeksan vuoden ajan (Gelbmann 2018). WordPress ei kuitenkaan sovi esimerkiksi verkkokauppa-tyyppisille projekteille, vaan parhaiten se sopii tietoa välittäville projekteille (Kauppinen 2015). WordPress kehitettiin alun perin bloggaustarkeitukseen ja se näkyy edelleen tietyissä ominaisuuksissa. WordPress on helppokäyttöinen ja sivuston saa nopeasti pystyyn, mutta lisäosilla sivustolle saadaan tehtyä monimutkaisiakin toimintoja. (Tuominen 2013.) Myös WordPress on rakennettu PHP ja MySQL-tekniologioiden päälle (W3Techs 2009-2018b).

Drupal sopii monimutkaisiin projekteihin ja sivuille, joilla käy paljon vierailijoita. Drupalilla tehdyt sivut ovat tehokkaita ja nopeita käyttää, mutta tekovaihe on hidas ja monimutkainen, mikä tarkoittaa myös korkeampaa hankintahintaa. (Tuominen 2013.) Drupalilla sivuja tehtäessä vaaditaan enemmän ohjelmointitaitoja kuin Joomlaa tai WordPressiä käytettäessä. Sillä saa kuitenkin tehtyä monimutkaisia toteutuksia ja sopii esimerkiksi julkishallinnon verkkosivuille ja verkkokauppoihin. (Kauppinen 2015.)

## 5 Verkkosivujen käyttäminen markkinoinnissa

Tämän päivän digitaalisessa maailmassa ihmiset olettavat löytävänsä yritykset myös verkosta. Tietoa tuotteista ja palveluista etsitään yhä enemmän Internetistä, jolloin yrityksen löydettävyyttä tätä kautta mahdollistaa näkyvyys ja potentiaalisia uusia asiakkaita yritykselle. (Yrityksen-perustaminen.net 2018.) Näin ollen verkkosivut ovat tärkeä osa yrityksen Internet-markkinointia. Markkinoinnilla tarkoitetaan toimintaa, jolla yritys pyrkii saamaan valitseman kohderyhmän ostamaan yrityksen tuotteita tai palveluita. Verkkosivut ovat osa Internet-markkinointia, jonka hyötynä on suuren asiakasmäärän tavoittaminen helposti ja nopeasti.

Yrityksen verkkosivut ovat yksi tärkeimmistä Internet-markkinoinnin muodoista. Sivujen sisältö tulee suunnitella oman kohderyhmän mukaan, jotta asiasta kiinnostuneet ihmiset eli mahdolliset asiakkaat saadaan tulemaan sivuille, viihtymään sivuilla ja tekemään ostopäätös. Verkkosivut vaativat kuitenkin mainontaa, jotta ne olisivat potentiaalisten asiakkaiden löydettävissä. (Liljeroos.) Tätä varten tarvitaan suunnitelma sivujen mainostamiseen, jota yrityksen on pystyttävä seuraamaan halutessaan tuloksia (Yrityksen-perustaminen.net 2018).

Jotta verkkosivusto voi toimia markkinointivälineenä, on sinne saatava ihmisiä. Tehokkain tapa tavoittaa ihmiset on hakukoneoptimointi eli SEO (Search Engine Optimization). Hakukoneoptimoinnin tarkoituksena on nostaa yrityksen sivusto paremmalle sijoitukselle hakukoneiden tuloksissa käyttämällä kuvaavia hakusanoja ja -lauseita. Suurin osa ihmisistä etsii tietoa uudesta yrityksestä tai haluamastaan tuotteesta tai palvelusta Googlesta. (Suomen Digimarkkinointi Oy 2018a.)

Verkkosivujen sisältöä voi optimoida usealla eri tavalla. Kannattaa aloittaa hakusanatutkimuksella, jolla selvitetään, millä sanoilla hakukoneissa tehdään hakuja liittyen yrityksen tuotteisiin tai palveluihin. Kun sopivat hakusanat ovat tiedossa, lisätään ne verkkosivujen sisältöön, jotta hakukoneet löytävät sivut. Hakusanat on hyvä lisätä seuraaviin kohtiin verkkosivuilla: HTML-koodin meta-tiedot (sivun otsikko ja meta-kuvaus), verkko-osoite, sisältöteksti ja otsikot sekä kuvat. (Suomen Digimarkkinointi Oy 2018b.)

Sivun otsikko eli title HTML:n meta-tiedoissa näkyy paitsi sivun otsikkona verkkosivulla myös hakukoneen hakutulossivulla. Otsikossa kannattaa käyttää hakusanaa, mieluiten vielä heti otsikon alussa. Meta-kuvaus eli meta-description ei varsinaisesti vaikuta hakutulosten sijoituksiin, mutta koska kuvaus näkyy hakutuloksissa, se on oiva paikka mainostaa

yritystä ja houkutella ihmisiä klikkaamaan linkkiä ja siirtymään yrityksen sivustolle. (Suomen Digimarkkinointi Oy 2018b.)

Sivun verkko-osoite vaikuttaa hakukoneiden tuloksissa sijoittumiseen ja siinä kannattaa pyrkiä käyttämään hakusanoja ja kertomaan, mitä sivusto sisältää (Suomen Digimarkkinointi Oy 2018b). Toisaalta verkko-osoite kannattaa pitää mahdollisimman lyhyenä ja ytimekkäänä.

Varsinkin h1-tason otsikoissa kannattaa myös hyödyntää hakusanoja, sillä ne vaikuttavat hakutulossijoituksiin. Title ja h1-tason otsikko kannattaa pitää samankaltaisina. Sisältötekstin tulee olla hyvää kieltä ilman kirjoitusvirheitä ja se kannattaa kirjoittaa hakusanojen ympärille syventäen aihetta. (Suomen Digimarkkinointi Oy 2018b.)

Kuvat kannattaa nimetä kuvaavilla nimillä, eli jos kuvassa on kukka, kuvan tiedostonimeksi kannattaa tallentaa esimerkiksi kukka.jpg, eikä 120717.jpg (Suomen Digimarkkinointi Oy 2018b).

Sivuston sisäisen optimoinnin lisäksi suurin ulkopuolella hakukoneisiin vaikuttava tekijä on ulkopuoliset linkit sivuille. Hakutuloksissa sijoittumiseen vaikuttavat sekä linkkien määrä että laatu. Saman aihepiirin toiselta sivustolta tuleva linkki omalle sivustolle on arvokkaampi kuin täysin aihepiiriin kuulumattomalta sivulta tuleva linkki. Jotkut linkit ovat myös arvostetumpia hakukoneissa kuin toiset. Esimerkiksi Helsingin Sanomien sivuilta tuleva linkki on arvokkaampi kuin yksityishenkilön sivuilta tuleva linkki. Muilla sivustoilla olevat linkit toimivat myös mainoksina yrityksille, sillä niiden kautta saadaan ihmisiä omalle sivustolle. (Liljeroos.)

Toinen tärkeä ulkopuolella tapahtuva toiminta on yrityksen sivujen jakaminen sosiaalisessa mediassa ja niiden keräämä huomio (Liljeroos).



## 6 Vaatimusmäärittely

Vaatimusmäärittelyllä määritetään, miksi projekti, eli tässä työssä sivusto, on tarpeellinen ja mitä tarpeita sillä tulee tyydyttää. Se keskittyy sivuston vaatimuksiin, ei sen tekniseen toimintaan. Vaatimukset voidaan jakaa toiminnallisiin ja ei-toiminnallisiin vaatimuksiin: Toiminnallisuus tarkoittaa sivuston toimintoja ja vaatimuksia ja ei-toiminnallisuudella tarkoitetaan suorituskyykyyn liittyviä asioita. (Kaskela 2005.)

Vaatimusmäärittelyprosessi voidaan jakaa kolmeen osaan: tavoitteiden määrittely, tarpeiden tunnistus sekä vaatimukset. Tavoitteiden määrittelyllä pyritään selvittämään, mitä verkkosivustolta halutaan ja miksi sen hankinta on tarpeellista? Ensin mietitään, mitä liike-toiminnallisia tavoitteita yrityksellä on, jonka jälkeen näiden tavoitteiden pohjalta voidaan määritellä konkreettiset tietotekniset tavoitteet. (Kaskela 2005.)

Tarpeiden tunnistus on vaatimusmäärittelyn toinen vaihe, jossa määritellään, mikä on lähtötilanne: Kehitetäänkö vanhaa vai luodaanko täysin uutta? Mikä on nykytilanne ja nykyprosessit sekä niiden ongelmat? On hyvä myös pohtia, miten sivusto vaikuttaa sidosryhmiin, kuten asiakkaisiin. Tarpeiden tunnistusta tukee ensimmäisen vaiheen tavoitteiden määrittely, mutta koska eri tahojen tarpeet voivat olla ristiriidassa toistensa kanssa, tulee tarpeet käydä tarkkaan läpi ennen siirtymistä kolmanteen vaiheeseen. (Kaskela 2005.)

Kolmas ja viimeinen vaihe vaatimusmäärittelyssä on tarpeiden perusteella määritellyt vaatimukset. Tässä vaiheessa pyritään hahmottamaan toimintakokonaisuus eli miten eri tarpeet yhdistetään, miten prosessi kokonaisuutena toimii ja mikä on sivuston rooli. Nykytilan ja nykyprosessien pohjalta mallinnetaan tavoitetila ja sen prosessit. On tärkeää myös priorisoida vaatimukset sivuston hyvän toimivuuden määrittelyä varten. (Kaskela 2005.)

Tiedon kerääminen on tärkeää koko vaatimusmäärittelyn aikana ja sitä voidaan kerätä esimerkiksi haastatteluilla sekä kyselyiden avulla. Tietoa voi hankkia myös jo olemassa olevista materiaaleista. (Kaskela 2005.)

Perinteisen vaatimusmäärittelyn ohelle on tullut myös toinen tapa määritellä vaatimuksia: käyttötapaukset. Käyttötapauksia voidaan kuvata joko sanallisesti tai kaavioina. Määrittely aloitetaan käyttäjien sekä roolien tunnistamisella. Käyttötapauksella voidaan kuvata, miten tietty käyttäjä käyttää sivustoa tietyssä tilanteessa. Käyttötapauksien määrittely sopii parhaiten prosessien kuvaamiseen ja niillä voidaan määritellä vain toiminnallisia vaatimuksia. (Kähönen 2016.)

## 7 Käyttäjäkokemus

ISO (International Organization for Standardization, kansainvälinen standardisointijärjestö) määrittelee käyttäjäkokemuksen ihmisen huomioiksi ja reaktioiksi liittyen tuotteen, järjestelmän tai palvelun käyttöön tai odotettuun käyttöön (ISO 2010).

Täydellisen kokonaiskäyttäjäkokemuksen pohjana on tuote tai palvelu, joka täyttää kaikki asiakkaan tarpeet helposti. Käyttäjäkokemus kattaa loppukäyttäjän kaiken vuorovaikutuksen yrityksen sekä sen palveluiden ja tuotteiden kanssa. Tuotteen tai palvelun tulee olla yksinkertainen, mutta tyylikäs, jotta asiakas haluaa mielellään käyttää sitä. Hyvän käyttäjäkokemuksen aikaansaamiseksi tarvitaan yrityksen eri toimintojen ja osastojen saumattomaa yhteistyötä. (Norman & Nielsen 1998-2018.)

Peter Morvillen hunajakennokuvio verkkosivuston käyttäjäkokemuksesta kertoo käyttäjäkokemuksen seitsemän osiota (kuva 3).



Kuva 3. Peter Morvillen hunajakennokuvaus verkkosivuston käyttäjäkokemuksesta (Morville 2004)

Morvillen hunajakennokuvio on rakennettu nimenomaan verkkosivuston suunnittelua silmällä pitäen ja hänen mukaansa käyttäjäkokemus rakentuu seitsemästä elementistä: arvo (valuable), hyödyllisyys (useful), käytettävyys (usable), haluttavuus (desirable), tavoitettavuus (findable), esteettömyys (accessible) sekä uskottavuus (credible). Käyttäjäkokemuksen ytimessä on sen varmistaminen, että yrityksen tarjoamalla tuotteella tai palvelulla, tai

tässä tapauksessa verkkosivustolla, on arvoa käyttäjälle. Yrityksen tulee varmistaa, että sivusto on hyödyllinen loppukäyttäjälle ja sitä on helppo käyttää. Sivusto tulee tehdä houkuttelevaksi ja sellaiseksi, että sieltä löytää helposti tietoa ja sen tulee olla saatavilla myös ihmisille, joilla on rajoitteita, kuten sokeille. Sivuston on myös elintärkeää luoda uskottava kuva yrityksestä ja tuotteesta. (Morville 2004.)

Käyttäjäkokemuksen suunnittelu on prosessi, jossa luodaan tuotteita, jotka tuottavat merkityksellisiä kokemuksia käyttäjälle. Se sisältää paitsi tuotteen käytettävyyden huolellisen suunnittelun myös nautinnon, jonka se tuottaa loppukäyttäjälleen. Siihen liittyy myös koko tuotteen hankinta- ja käyttöönottoprosessi, kuten käytettävyys, toiminnallisuus ja ulkonäkö. (Interaction Design Foundation.)

Käyttäjäkokemuksen suunnittelun tärkeitä puolia on ymmärtää, miten käyttäjäkokemus muodostuu. Ensikosketus tuotteeseen kehittyy ajan myötä, kun tuote on käytössä. Käyttäjän huomiot, toiminta ja ymmärrys muodostavat mieleenpainuvan ja yhtenäisen tarinan eli käyttäjäkokemuksen. Tunnereaktiot prosessin aikana määrittävät sen, tuleeko kokemuksesta positiivinen vai negatiivinen. (Interaction Design Foundation.)

Käyttäjäkokemuksen suunnittelijat pyrkivät tietoisesti luomaan ja muokkaamaan tekijöitä, jotka vaikuttavat tähän prosessiin, miettimällä tuotteen käyttöä kolmella kysymyksellä: miksi, mitä ja miten. Miksi käyttäjä haluaa hankkia tuotteen: tiettyä tekemistä varten vai siksi, että kyseisen tuotteen omistaminen ja käyttö herättävät tiettyjä ajatuksia? Mitä tuotteella voi tehdä, mitä toiminnallisuutta sillä on ja miten toiminnallisuus on toteutettu esteettömyyttä ja ulkonäköä ajatellen? (Interaction Design Foundation.)

Käyttäjäkokemuksen seitsemän eri osion on kaikkien toimittava, jotta käyttäjäkokemuksesta tulisi hyvä. Jos yksi tai useampi osio ei toimi, vaikuttaa se kokonaiskäyttäjäkokemukseen. (Derome 2015.) Oletetaan esimerkiksi, että etsit remontointiyritystä Internetistä ja löydät yrityksen X kotisivut helposti ja alat selailla niitä. Sivusto on hyvin suunniteltu, se on kaunis ja yrityksellä on paljon kuvia jo tehdyistä töistä. Yritys vaikuttaa hyvältä ja kokemukseesi tähän mennessä on hyvä. Päätät ottaa yritykseen yhteyttä keskustellaksesi remontoinnista, mutta et meinaa millään löytää yrityksen yhteystietoja. Kun vihdoinkin löydät ne, tajuat, ettei itse yrityksestä ole kerrottu sivustolla oikeastaan yhtään mitään. Tämä vaikuttaa kokemukseesi ja alat miettiä, kuinka luotettava yritys oikeastaan on.

## 8 Benchmarking

Benchmarking on vertailuanalyysia ja vertaiskehittämistä. Se tarkoittaa toimintaa, jossa yrityksen toimintaa ja prosesseja vertaillaan muihin yrityksiin, jotka ovat yleensä paremmin menestyviä kuin oma yritys ja ne voivat olla miltä tahansa toimialalta. Benchmarking-menetelmän avulla voidaan kehittää oman yrityksen toimintaa oppimalla sekä oman yrityksen heikkoja puolia että muiden yritysten vahvoja puolia. Menetelmää voidaan toteuttaa esimerkiksi tutustumalla yritykseen paikan päällä tai etsimällä tietoa kirjoista, lehdistä ja Internetistä. (Itä-Suomen yliopisto.)

Benchmarkingia voi käyttää havainnoimaan monimutkaisia yritysrakenteita tai tunnistamaan mahdollisia kehitysalueita. Benchmarkingille ominaista on etsiä järjestelmällisesti tehokkaita käytäntöjä tai parempia ratkaisuja monimutkaisiin ongelmiin ja prosesseihin. Etsintä keskittyy muiden kuin oman alan yritysten analysointiin. Menetelmä ei pyri löytämään eroja yritysten välillä, vaan parhaimmat käytännöt, jotka auttavat pitämään yllä yrityksen parempaa kilpailuetua. (Global Benchmarking Network 2018a.)

Benchmarkingia on harrastettu jo pitkään. Ensimmäinen tunnettu tutkimus aiheesta on 1900-luvun alusta, kun Saksan armeija tutustui amerikkalaisen sirkuksen käytäntöihin siirtää koko sirkus tavaroineen, eläimineen ja ihmisineen toiseen kaupunkiin yhdessä yössä. Tutustumisen ansiosta Saksan armeija pystyi kehittämään tapojaan siirtää joukot toiseen sijaintiin. Toinen varhainen esimerkki benchmarkingin käytöstä on vuodelta 1916, kun Henry Ford toi kokoonpanolinjat autoteollisuuteen. Ford oli käynyt tutustumassa teurastamoon Chicagossa, jossa ruhonosia siirrettiin työpisteestä toiseen riippukiskojen avulla. (Global Benchmarking Network 2018b.)

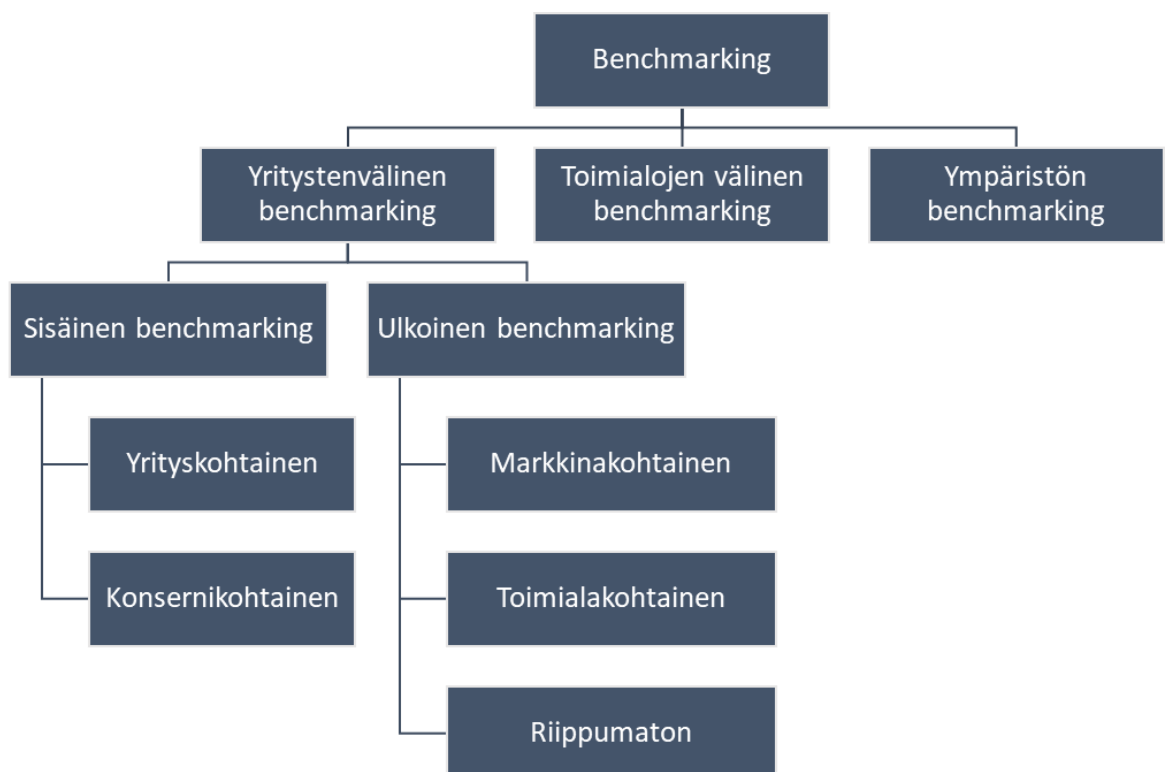
Benchmarking-termin keksi Xerox vuonna 1979, kun he huomasivat Japanin myyvän kopiokoneita paljon halvemmalla kuin mitä Xeroxin tuotantokustannukset olivat. Xerox teki tuotanto-osastollaan benchmarking-tutkimuksen, jossa kaikkien myynnissä olevien kopiokoneiden tuotantokustannukset ja ominaisuudet analysoitiin. Menetelmä oli menestys, minkä ansiosta Xerox toteutti benchmarking-tutkimukset kaikilla liiketoiminta-alueillaan. (Global Benchmarking Network 2018b.)

Benchmarking-menetelmän hyötyjä on sekä suoria että epäsuoria. Suoria hyötyjä ovat yrityksen tarkka analysointi, yrityksen tai sen osien vertailu, parhaiden käytäntöjen määrittely, toimintakyvyn heikkouksien tunnistaminen sekä vaihtoehtoisten ratkaisujen arviointi. Epäsuoria hyötyjä ovat oman yrityksen liiketoiminnan prosessien ymmärtämisen edistämi-

nen, yrityksen tavoitteiden kyseenalaistaminen, yritysstrategian varmentaminen, kilpailu-  
aseman vahvistaminen sekä jatkuvan kehityksen prosessin aloittaminen. (Global  
Benchmarking Network 2018c.)

Benchmarkingin voi jakaa kolmeen eri päätyyppiin (kuva 4): yritystenvälinen vertailu, toi-  
mialojen välinen vertailu sekä ympäristön vertailu. Yritystenvälinen vertailu on kaikista laa-  
jimalle levinnyt benchmarking-menetelmä, jossa yritykset oppivat toisiltaan vertailemalla  
ja vaihtamalla tietoa tutkinnan kohteena olevista aiheista. Toimialojen välisessä vertai-  
lussa vertaillaan yksittäisten toimialojen suorituksia ja pyritään oppimaan muilta paremmin  
toimivilta toimialoilta. Ympäristön vertailumenetelmällä maat voivat tulevaisuudessa ver-  
tailla poliittisia, yhteiskunnallisia ja taloudellisia ympäristöjä ja oppia toisiltaan. (Global  
Benchmarking Network 2018d.)

Yritystenvälisen benchmarkingin voi jakaa vielä sisäiseen ja ulkoiseen benchmarkingiin ja  
ne yrityskohtaiseen ja konsernikohtaiseen (sisäinen benchmarking) sekä markkinakohtai-  
seen, toimialakohtaiseen ja riippumattomaan (ulkoinen benchmarking) (Global Benchmar-  
king Network 2018d). Jako on kuvattu kuvassa 4.



Kuva 4. Benchmarkingin tyypit (Global Benchmarking Network 2018d)

Sisäinen benchmarking on menetelmän yksinkertaisin muoto, jossa yritys pyrkii oppimaan omista käytännöistään. Sisäistä benchmarkingia voidaan käyttää menetelmään tutustumiseen ja pohjana ulkoisissa benchmarking-tutkimuksissa. Sisäisen benchmarkingin voi jakaa yrityskohtaiseen ja konsernikohtaiseen vertailuun. Yrityskohtainen benchmarking on haastavaa, sillä yrityksen sisäisissä prosesseissa on esimerkiksi teknisiä ominaisuuksia, joita on vaikea vertailla muihin prosesseihin. Konsernikohtaisessa vertailussa vertaillaan konsernin eri osia tai yhtiöitä toisiinsa heikkouksien ja parhaiden käytäntöjen löytämiseksi. (Global Benchmarking Network 2018e.)

Ulkoisessa benchmarkingissa yrityksen käytäntöjä vertaillaan muiden yritysten käytäntöihin. Jotta yritys olisi vertailukelpoinen toisen yrityksen kanssa, on ensin tunnistettava yhtiöläisyydet yrityksissä. Ulkoisen benchmarkingin voi jakaa vielä markkinakohtaiseen, toimialakohtaiseen ja riippumattomaan vertailuun. Markkinakohtainen vertailu on vertailua kilpailevien yritysten välillä, jossa tietoa vaihdetaan suoraan yritysten välillä ja opitaan yhdessä. Haasteita menetelmän toteuttamiseen aiheuttaa yritysten haluttomuus jakaa tietoa kilpailijoiden kanssa. Toimialakohtainen vertailu on laajempaa kuin kahden yksittäisen yrityksen vertailu. Se keskittyy trendeihin ja tutkii tietyn toiminnon tehokkuutta koko toimialalla. Toimialakohtaisessa vertailussa on välttämätöntä tutkia useampia yrityksiä kuin markkinakohtaisessa vertailussa. Toimialakohtaisen vertailun etuja verrattuna markkinakohtaiseen vertailuun on kilpailuaseman puuttuminen: vertailun kohteena olevat yritykset toimivat eri markkinoilla. Näin tiedon jakaminen on helpompaa. Riippumattomassa vertailussa vertailun kohteeksi haetaan ”parasta” yritystä toimialasta riippumatta ja pyritään oppimaan heiltä parhaita käytäntöjä, jotka muokataan omaan yritykseen sopiviksi. (Global Benchmarking Network 2018f.)

## 9 Kohderyhmän kartoitus ja määrittäminen

Segmentointi eli kohderyhmän määrittely on tärkeä osa yrityksen markkinointia. Sen tarkoituksena on löytää yrityksen tärkeimmät asiakasryhmät eli ne tahot, jotka eniten tarvitsevat yrityksen palveluita tai saavat niistä suurimman hyödyn. Yrityksen edun mukaista ei olisi tarjota palveluita kaikille, vaan rajaamalla asiakasryhmänsä pienemmäksi, pystyy yritys vastaamaan asiakkaidensa tarpeisiin paremmin. Yksi tärkeimmistä segmentoinnin tavoitteista on erottuminen kilpailijoista sekä sen varmistaminen, että asiakkaat valitsisivat juuri kyseisen yrityksen palvelut. (Bergström & Leppänen 2009, 150.)

Segmentointi aloitetaan yrityksen asiakasryhmien tunnistamisella. Yritys jaottelee asiakkaansa eri ryhmiin ostokäyttäytymisen perusteella: mitä asiakas ostaa ja miksi. Kun asiakkaat on jaettu eri ryhmiin, valitsee yritys niistä itsellensä tärkeimmät eli ne, jotka ovat yrityksen toiminnan kannalta kannattavimmat. Useimmiten tehokkainta voi olla valita asiakasryhmistä vain yksi eli kaikista tärkein. Näin yritys pystyy keskittämään toimintansa ja palvelunsa vastaamaan vain ja ainoastaan kyseisen asiakasryhmän tarpeita. (Bergström & Leppänen 2009, 151.)

Myös yrityksen kotisivujen suunnittelussa on tärkeää heti alkuun perehtyä siihen, mikä on sivuston tavoite sekä pääasiallinen kohderyhmä. Kyseisten tietojen pohjalta sivuston kehitystyö lähtee varmimmin etenemään haluttuun suuntaan, koska tietoja pystytään hyödyntämään valittaessa sivuille sisältöä sekä suunniteltaessa sivujen yleisilmettä. On tärkeää, että sivut palvelevat sille asetettuja tavoitteita ja valittua kohderyhmää mahdollisimman hyvin. Näin ollen yrityksen tulee tarkasti miettiä valitsemansa kohderyhmän tarpeita ja haluja ja tämän pohjalta suunnitella sivuston sisältöön ja ulkonäköön liittyviä seikkoja, jotka palvelisivat haluttua kohderyhmää. (Fulmore 2011.)

Segmentointiperusteet voidaan jakaa kahteen ryhmään: koviin ja pehmeisiin segmentointiperusteisiin. Kovat perusteet ovat kiinteitä ja muuttumattomia ja niihin lasketaan esimerkiksi ikä ja toimipaikka. Pehmeät perusteet ovat häilyväisempiä ja valinnaisia ja niitä on helpompi muuttaa. Pehmeisiin perusteisiin lukeutuvat esimerkiksi harrastukset ja mieltymykset. Yrityksen asiakasrekisteriä voidaan käyttää segmentoinnin pohjana ja tietoa voi syventää esimerkiksi kyselytutkimuksilla, joilla selvitetään asiakkaiden ostokäyttäytymistä, arvoja ja asenteita. Segmentointia voi käyttää sekä nykyisiin että mahdollisiin asiakkaisiin ja ryhmitellä asiakkaita esimerkiksi tuotteiden käytön tai asenteiden perusteella. Mitä pienempiin ryhmiin asiakkaat pystyy jakamaan, sen helpompi on suunnitella ja kohdentaa tiettylle ryhmälle sopivaa markkinointia. (Republica.fi.)

Kohderyhmän määrittely voidaan aloittaa valitsemalla valmis segmentti, esimerkiksi alueen, sukupuolen tai kiinnostusten perusteella, jolla saadaan mahdollisesti ihmiset kiinnostumaan omasta yrityksestä ja tuotteesta tai palvelusta, mutta valmiin segmentin lisäksi täytyy pohtia, miten kiinnostuneet ihmiset saataisiin myös ostamaan (Ansaharju 2011).



## 10 Käyttöohjeiden tuottaminen

Hyvä käyttöohje on helppolukuinen ja ymmärrettävä. Ohjeessa tulee käyttää mahdollisuuksien mukaan yleiskieltä ja jos se ei ole mahdollista, tulee erikoistermit selittää. Lau-seiden tulee olla lyhyitä ja yksinkertaisia ja ohjeen tulee sisältää vain oleellista käyttöön liittyvää tietoa. Kuvien, värien, selkeän asettelun sekä erilaisten kirjainkokojen ja -tyylien käyttäminen voi parantaa ohjetta. (Tukes 2016.)

Alkuperäisen käyttöohjeen laatijan on hyvä olla se, joka on tuotteen tehnyt, koska hän tietää parhaiten tuotteen käyttötarkoituksen ja kaikki ominaisuudet (Tukes 2016). Käyttöohjeen laadinnassa kannattaa pitää mielessä silmäiltävyys, kieli ja luettavuus, helppokäyttöisyys sekä juonellisuus (VirtuaaliAMK a).

Silmäiltävyys tarkoittaa sitä, kuinka helposti ja nopeasti lukija pystyy hahmottamaan tekstin kokonaisuutena. Sitä voi parantaa hyvällä otsikoinnilla, sopivan pituisilla kappaleilla sekä luettelilla. Silmäiltävyyttä parantaa myös hyvä tekstinasettelu, kuten sopiva riviväli, kirjainkoko ja marginaalit. Väriyksen tulisi olla yhtenäinen ja käyttötarkoitukseen sopiva. (VirtuaaliAMK b.)

Kielen tulee olla ymmärrettävää ja tekstin helppolukuista. Lukukokemusta voi parantaa käyttämällä yleiskieltä ja selkeitä virkerakenteita sekä havainnollistamalla tekstiä esimerkiksi taulukoilla ja kuvilla. Kielen tulee olla asiallista ja verbimuodoissa kannattaa suosia aktiivimuotoja passiivin sijaan. (VirtuaaliAMK c.)

Helppokäyttöisyydellä tarkoitetaan sitä, kuinka helppo lukijan on löytää tarvitsemansa tieto ohjeesta. Sitä voi parantaa esimerkiksi hyvällä sisällysluettelolla ja kuvaavilla otsikoilla. Hyvä silmäiltävyys ja luettavuus parantavat ohjeen käytettävyyttä. (VirtuaaliAMK d.)

Juonellisuus auttaa myös ohjeissa. Se tarkoittaa sitä, että ohje etenee loogisesti ja auttaa lukijaa asioiden toisiinsa yhdistämisessä. Ohjeen juoni näkyy syy-seuraussuhteen esittämisessä, esimerkeissä ja vertauksissa. (VirtuaaliAMK e.)

Ennen ohjeen varsinaista laatimista tulee koota materiaali ohjetta varten sekä suunnitella ohjeen sisältö ja esitystapa. Ennen käyttöohjeen luovuttamista tulee tarkistaa, että kieli on ymmärrettävää, eikä siinä ole kirjoitusvirheitä. Ohje tulee myös testata sekä korjata, jos testauksessa ilmenee virheitä tai epäjohtonmukaisuuksia. (VirtuaaliAMK f.)

Jos aihe on hyvin laaja, voi olla järkevää tuottaa sekä laaja käyttöopas, jossa käydään läpi yksityiskohtaisesti kaikki mahdollinen, että lyhyempi pikaopas, jota voi käyttää tukena, kun aihe on jo jotenkin hallussa.

## 11 MCV Remonttipalvelun verkkosivusto

Opinnäytetyön empiirisen osan tuotoksena syntyi verkkosivusto. Toimeksiantajalla oli tarve saada yritykselleen sivusto, jolla he voisivat esitellä tehtyjä projekteja. Asiakas ei halunnut kovin monimutkaista sivustoa, joten siitä päätettiin tehdä yksisivuinen osioihin jaettu sivusto. Sivuston suunnittelu alkoi asiakkaan kanssa käydylä palaverilla, jonka keskustelujen pohjalta tehtiin prototyyppi sivustosta. Prototyypille tehtiin think aloud -käytettävyydesti, jonka jälkeen luotiin Adobe Photoshopilla sivustosta visuaalinen suunnitelma ja ohjelmoitiin sitä ja prototyyppiä mallina käyttäen sivusto. Lopulta sivustolle tehtiin vielä uusi think aloud -käytettävyydesti, jotta varmistuttiin, että sivusto on selkeä ja toimiva.

### 11.1 Yrityksen esittely

Projektin toimeksiantaja oli Vantaalla sijaitseva remontointialan pienyritys MCV Remonttipalvelu ja Kotiapu. Yritys tekee erilaisia remonttitoita pääkaupunkiseudun alueella, lähinnä kylpyhuoneremontteja. He ovat erikoistuneet mikrosementointiin, joka on tekniikka, jolla saa modernin ilmeen pinnoille.

Yritys tarjoaa myös esimerkiksi siivouspalveluita, johon viittaa yrityksen nimessä oleva ”Kotiapu”, mutta asiakas halusi sivustollaan mainostaa ainoastaan remontointia, eikä mitään ”Kotiapuun” liittyvää. Koska yrityksen logossa oli kuitenkin sana ”Kotiapu”, päätimme yhdessä asiakkaan kanssa poistaa logosta sanan ”Kotiapu”, jottei se hämää sivustolla vieraillevia kävijöitä. Päätös oli oikea, sillä prototyypille tehdyssä think aloud -käytettävyydesti, jossa logossa vielä näkyi sana ”Kotiapu”, testihenkilöt ihmettelivät sanan merkitystä.

### 11.2 Projektisuunnitelma

Projekti suunniteltiin toteutettavaksi noin viiden kuukauden aikana ja projektisuunnitelma aikataulutettiin 21 viikolle (liite 2). Projekti alkoi syksyllä 2017 ja tavoite oli saada sivusto ja siihen liittyvä raportointi valmiiksi alkukeväältä 2018. Projekti jakaantui neljään osioon: alkuvalmistelut ja keskustelut asiakkaan kanssa, sivuston visuaalisen ilmeen suunnittelu, sivuston ohjelmointi sekä viimeistely ja sivuston luovuttaminen asiakkaalle.

Projekti alkoi palaverilla asiakkaan kanssa, jossa kartoitettiin asiakkaan tarpeet sekä tavoiteltava kohderyhmä. Seuraavaksi analysoitiin vastaavia verkkosivustoja ja hyödynnettiin tuloksia suunniteltavalla verkkosivustolla. Seuraavassa vaiheessa suunniteltiin sivusto ja tehtiin siitä prototyyppi, tehtiin sille think aloud -käytettävyydesti ja testin perusteella korjattiin prototyyppiä. Tämän jälkeen prototyypin pohjalta tehtiin visuaalinen suunnitelma

ja näiden kahden avulla aloitettiin itse sivuston ohjelmointi, johon oli varattu projektin kokonaisajasta noin puolet. Kun sivusto oli ohjelmoitu, tehtiin sille uusi think aloud -käytettävyydestesti, jonka jälkeen sivusto luovutettiin asiakkaalle.

### **11.3 Sivuston toteutustavan valintaperuste**

Sivuston toteutustapaa mietittiin pitkään. Alkuperäinen suunnitelma oli ohjelmoida sivusto alusta lähtien itse, mutta harkinnassa oli myös WordPress. WordPress oli varteenotettava vaihtoehto ja olisi tarjonnut monia hyötyjä, mutta päätös tehtiin palata alkuperäiseen suunnitelmaan ohjelmoida sivusto kokonaan itse eri teknologioita hyödyntäen, koska sivustosta haluttiin tietyn näköinen ja mahdollisimman lähelle asiakkaalla hyväksyttyä prototyyppiä.

Tärkein syy toteutustavan valintaan oli se, että saatiin varmistettua, että sivuston rakenteesta ja tyylistä saatiin juuri sellainen kuin oli tarkoituskin. Tavoitteena oli myös oppia ohjelmoinnista ja eri tekniikoista ja toisaalta näyttää jo olemassa olevaa osaamista ohjelmoinnin suhteen.

### **11.4 Käyttöohjeistus asiakkaalle**

Sivusto toteutettiin alusta lähtien ohjelmoimalla, ja asiakkaan on tarkoitus itse päivittää sivustoa. Tarve päivittämiselle koskee vain kuvien päivittämistä, joten asiakkaalle tuotettiin kirjalliset ohjeet, joita seuraamalla he pystyvät itse päivittämään sivuston kuvagalleriaan uusia kuvia. Kaikki muut tarvittavat päivitykset sivustoon sovittiin tehtäväksi työn tekijöiden (Eklund/Khamaiseh) toimesta.

Käyttöohjeet kattavat ilmaisen kuvankäsittelyohjelma GIMP:n asennuksen koneelle sekä kuvien koon muokkaamisen GIMPillä tarvittaviin kahteen kokoon (pieni kuva yleisnäkymään ja iso kuva avatulle kuvalle). GIMP:n käyttöohjeet ovat liitteissä 3 ja 4. Lisäksi ohjeissa käydään läpi tekstieditori Notepad++:n asentaminen koneelle ja tiedoston käsittely ohjelmalla. Notepad++ -ohjeet ovat liitteissä 5 ja 6. Liite 7 sisältää käyttöohjeen koodin muokkaamiseen, jotta uudet kuvat päivittyvät sivuston kuvagalleriaan. Kaikki käyttöohjeet testattiin ja todettiin selkeiksi ja toimiviksi.

### **11.5 Toteutus**

Projekti aloitettiin tutustumalla muutamien remointipalveluita tuottavien yritysten verkkosivustoihin. Tämän tarkoituksena oli ottaa selvää, millaisia sivustoja asiakkaan mahdollisilla kilpailijoilla on. Kilpailijoiden sivustoja pyrittiin hyödyntämään projektissa siten, että

niistä poimittiin niin hyviä kuin huonojakin ominaisuuksia, joiden pohjalta asiakkaan sivustoa lähdettiin suunnittelemaan.

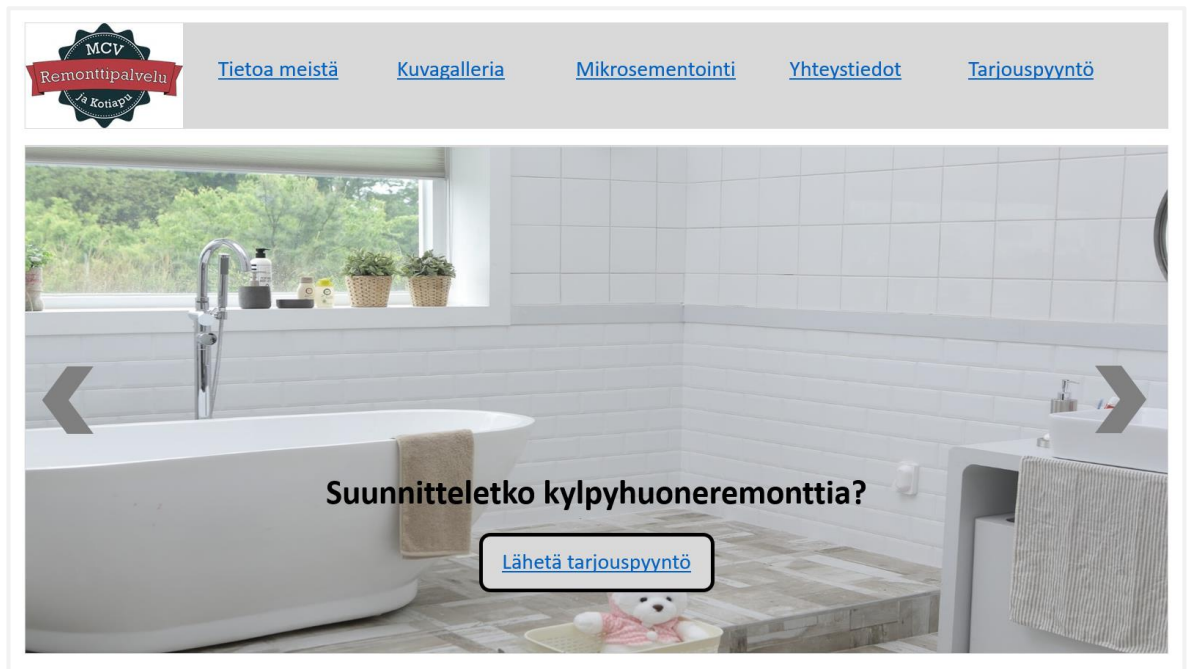
Kilpailijoiden sivuja läpikäydessä muodostui melko selkeä kuva siitä, millaisia asioita ja ominaisuuksia hyvän sivuston tulisi pitää sisällään. Tämän lisäksi haastattelimme useita henkilöitä siitä, millaisia asioita he toivoisivat löytävänsä remontointiyrityksen verkkosivuilta, jos he itse olisivat tilanteessa, jossa tarvitsisivat kyseisiä palveluita. Kilpailijoiden sivustojen läpikäynti sekä haastattelujen tulokset yhdessä antoivat hyvän pohjan asiakkaan sivuston suunnitteluun. Näiden pohjalta asiakkaalle pystyttiin esittelemään idea sivustosta ja sen tärkeimmästä sisällöstä ja ominaisuuksista.

Keskustelut asiakkaan kanssa johtivat kuitenkin siihen lopputulokseen, ettei heillä ollut kiinnostusta ottaa sivuille moniakaan niistä ominaisuuksista, jotka alustavassa tutkimuksessa olivat osoittautuneet tärkeiksi. Asiakas ei halunnut yrityksensä sivuista erityisen markkinoivia, vaan heille oli tärkeintä, että sivut toimisivat ikään kuin portfoliona, jonne mahdollisia asiakkaita voisi ohjata katsomaan referenssikuvia yrityksen aiemmin tekemistä remonteista. Näin ollen ensimmäistä prototyyppiä lähdettiin rakentamaan täysin asiakkaan toivomien ominaisuuksien pohjalta.

### **11.5.1 Prototyyppi**

Prototyyppi tehtiin PowerPoint-työkalua hyödyntäen (kuvat 5-13). Prototyypin tarkoituksena oli paitsi tehdä konkreettinen suunnitelma, minkälaista sivustoa aletaan rakentaa, myös toimia materiaalina käytettävyydestiä varten. Prototyyppi toimi mallina sivuston visuaalista suunnitelmaa tehdessä sekä sivuston ohjelmointivaiheessa.

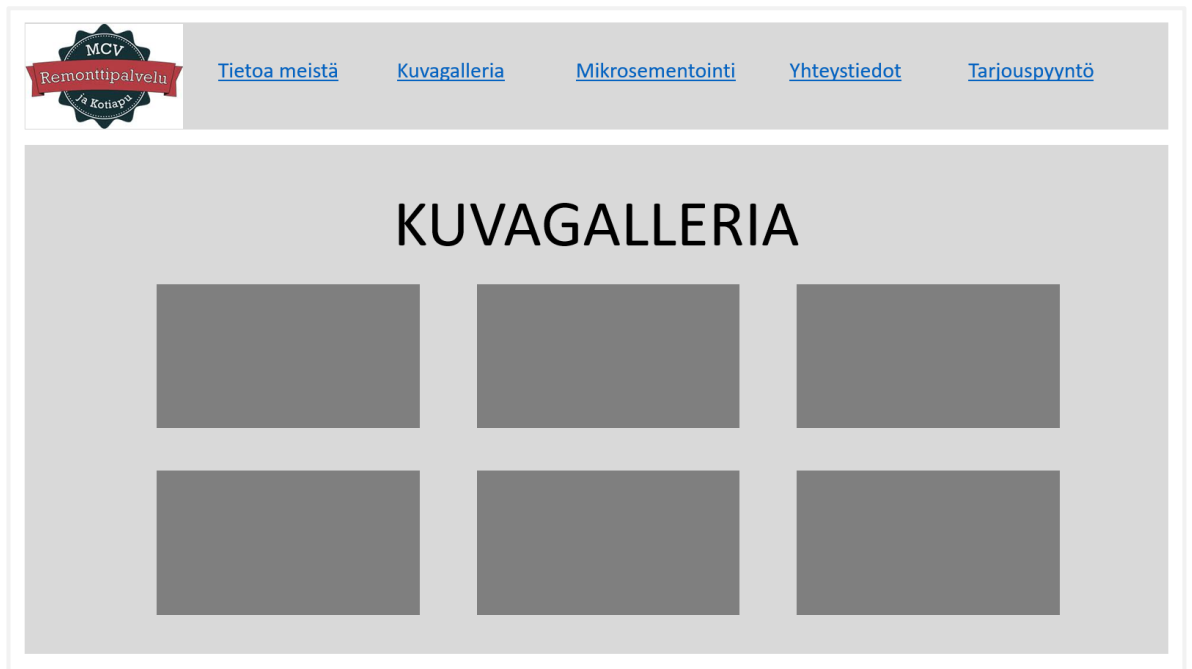
PowerPointin avulla prototyypistä saatiin mahdollisimman aidon oloinen sisällyttämällä siihen linkkien toimivuus, kuten oikeassa sivustossa. Tämän takia se oli helppo laittaa tietokoneen näytöllä testaajien eteen, koska he pystyivät liikkumaan prototyypin sivuilla aidon sivuston kaltaisesti. Näin prototyypin testaukseen saatiin mahdollisimman paljon aidon sivuston toimintaa ja tätä kautta hyviä kommentteja lopulliseen sivustoon liittyen.



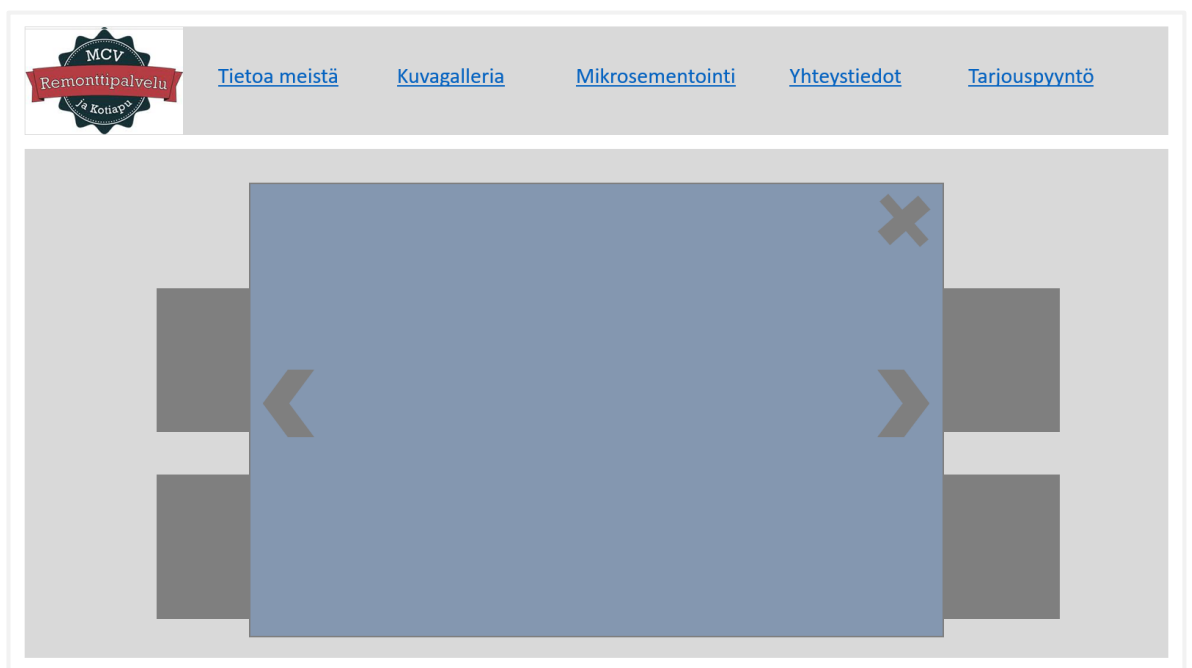
Kuva 5. Prototyypin etusivu



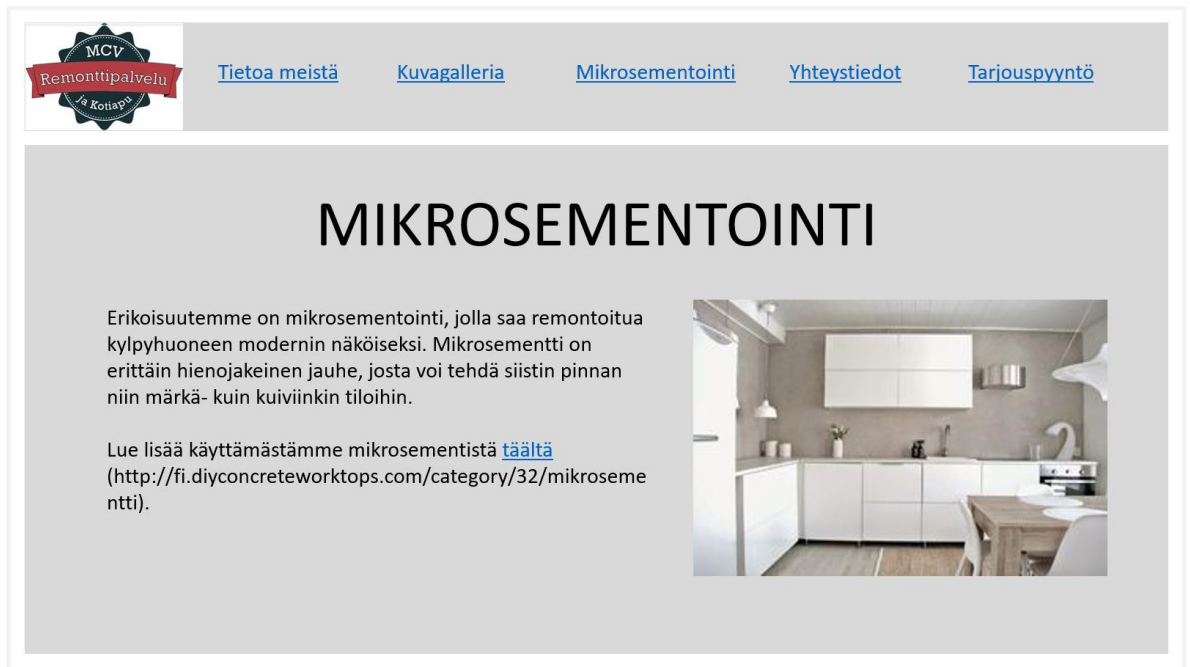
Kuva 6. Prototyypin "Tietoa meistä" -sivu



Kuva 7. Prototyypin "Kuvagalleria"-sivu



Kuva 8. Prototyypin kuvagallerian avattu kuva



Kuva 9. Prototyypin "Mikrosementointi"-sivu



Kuva 10. Prototyypin "Yhteystiedot"-sivu





[Tietoa meistä](#)
[Kuvagalleria](#)
[Mikrosementointi](#)
[Yhteystiedot](#)
[Tarjouspyyntö](#)

Ota yhteyttä oheisella lomakkeella, mikäli haluat lisätietoja tai tehdä tarjouspyynnön.

Nimi


Osoite

Sähköposti

Puhelin

☐ Tarjouspyyntö  
☐ Tiedustelu

Kuva 11. Prototyypin lomakesivu – valinta



[Tietoa meistä](#)
[Kuvagalleria](#)
[Mikrosementointi](#)
[Yhteystiedot](#)
[Tarjouspyyntö](#)

Ota yhteyttä oheisella lomakkeella, mikäli haluat lisätietoja tai tehdä tarjouspyynnön.

Nimi

Osoite

Sähköposti

Puhelin

Kylpyhuoneen pinta-ala

Halutut työt

**LÄHETÄ**

Kuva 12. Prototyypin lomakesivu – tarjouspyyntö



[Tietoa meistä](#)
[Kuvagalleria](#)
[Mikrosementointi](#)
[Yhteystiedot](#)
[Tarjouspyyntö](#)

Ota yhteyttä oheisella lomakkeella, mikäli haluat lisätietoja tai tehdä tarjouspyynnön.

Nimi

Osoite

Sähköposti

Puhelin

Lisätietoja

LÄHETÄ

Kuva 13. Prototyypin lomakesivu – tiedustelu

### 11.5.2 Think aloud -käytettävyystesti

Think aloud -käytettävyystesti suoritettiin sekä prototyypille että valmiille sivustolle. Kukin testiin osallistunut henkilö teki testin itsenäisesti valvojan (Eklund/Khamaiseh) seurattessa vieressä. Testi suoritettiin tietokoneella. Testaajaa ohjeistettiin tutustumaan ja liikkumaan sivustolla ensin itsenäisesti ja tekemään huomioita ja ehdotuksia liittyen sivuston toiminnallisuuteen, sisältöön sekä ulkonäköön, jonka jälkeen testaajalle annettiin tehtäviä suoritettavaksi sivustolla.

#### Prototyypin think aloud -käytettävyystesti

Prototyypille tehtyyn käytettävyystestiin osallistui seitsemän (7) henkilöä. Testaajien ikäkauma oli 25–69 vuotta ja testaajina oli sekä miehiä että naisia useilta eri toimialoilta. Tietokoneen käytön osaamisen hallinta oli eri tasoilla. Testihenkilöillä nousi paljon samankaltaisia huomioita ja ehdotuksia esiin, varsinkin sivuston värimaailmaan liittyen.

Huomiot prototyypin toiminnallisuudesta ja sisällöstä:

- Linkki etusivulle puuttuu.
- Kun on valinnut Tarjouspyyntö-sivulla esim. tarjouspyyntö-valinnan, voisi aukeavaan näkymään lisätä Peruuta-napin, jotta voi palata valitsemaan tiedustelun.
- Lomakkeelta puuttuu selitys, mitä tiedustelu tarkoittaa.
- Usein kysytyt kysymykset -lista olisi hyvä.
- Tiedustelu- ja tarjouspyyntö-lomakkeet voisivat olla erillään.
- Tarjouspyynnön tekeminen on selkeää.
- Olisi hyvä nähdä navigointipalkista, millä sivulla ollaan.
- Yhteystiedot ovat selkeästi merkitty.

Huomiot prototyypin ulkoasusta:

- Ulkoasu on liian pelkistetty, kaipaa väriä.
- Ulkoasu on selkeä.
- Taustakuvioksi voisi laittaa esimerkiksi mikrosementointi-kuviota.
- Mitä ”Kotiapu” tarkoittaa logossa?

Testaajien mielestä sivustolla olisi hyvä olla

- referenssejä
- selkeä esittely siitä, mitä palveluita yritys tarjoaa
- yritysesittely ja yhteystiedot
- kuvia tehdyistä töistä
- tietoa siitä, minkälaiset vakuutukset yrityksellä on ja mikä on takuu työlle
- tietoa, onko yrityksellä yhteistyökumppaneita/alihankkijoita ja jos on, miten työt jakaantuvat
- tieto siitä, mikä on palvelualue
- hintaesimerkkejä (onko käytössä urakka- vai tuntiveloitus)
- yhteydenottolomake.

Testaajia pyydettiin myös suorittamaan tehtäviä, minkä tavoitteena oli selvittää, onko sivusto helppokäyttöinen. Annetut tehtävät onnistuivat helposti kaikilta testaajilta.

Testaajille annetut tehtävät:

- Lähetä tarjouspyyntö.
- Tarkastele yrityksen esimerkkikuvia ja suurennä kuva paremmin näkyväksi.

Prototyypin think aloud -käytettävyydestin tulosten mukaisesti navigointipalkkiin lisättiin linkki takaisin etusivulle eli sivuston yläreunan kuvakaruselliin logoa klikkaamalla. Ulkoasuun tuotiin eloa lisäämällä sivulle taustakuvia sekä punaista väriä, jolla samalla yhdistettiin logo sivustoon. Sivuston jatkokehityksessä lomaketta kuitenkin yksinkertaistettiin, joten valintapainike tarjouspyynnölle ja tiedustelulle poistettiin ja sama lomake tehtiin palvelemaan sekä tarjouspyyntöä että muuta yhteydenottoa. Sivuston yksinkertaisen luonteen takia usein kysytyjen kysymysten lista jäi myös pois ja, koska jokaisella sivun osiolla on sama otsikko kuin navigointipalkissa, ei navigointipalkkiin myöskään lisätty merkkiä siitä, millä ”sivulla” ollaan. ”Kotiapu”-sana poistettiin logosta, koska asiakas ei halunnut sivuston mainostavan sitä osaa yrityksestä, vaan ainoastaan remontointipuolta.

Asiakkaan kanssa käytyjen keskustelujen pohjalta sivustolle ei myöskään lisätty kaikkia testaajien mielestä sinne kuuluvia asioita, kuten hintaesimerkkejä ja takuutietoja, koska sivuston haluttiin olevan pääasiallisesti portfolio yritykselle.

### **Valmiin sivuston think aloud -käytettävyydesti**

Valmiille sivustolle tehtyyn käytettävyydestiin osallistui viisi (5) ihmistä. Testaajien ikäkauma oli 31–70 vuotta ja testaajina oli sekä miehiä että naisia useilta eri toimialoilta. Tietokoneen käytön osaamisen hallinta oli eri tasoilla. Kaikki käyttäjät pitivät sivustoa selkeänä ja helppokäyttöisenä, ja annetut tehtävät onnistuivat kaikilta hyvin.

Testaajien mielestä sivusto on

- tyylikäs
- hieno
- helppokäyttöinen
- nykyaikainen
- asiallinen
- kiinnostusta herättävä.

Testaajat kehuivat myös sivuston alussa olevan kuvakarusellin kuvia, jotka luovat heidän mielestään tunnelmallisuutta sivustolle. Myös kuvagallerian toimivuus sai kiitosta. Erityisen paljon pidettiin siitä, etteivät kuvat aukea erilliseen ikkunaan, ja kun yhden kuvan avaa, pääsee samasta näkymästä siirtymään suoraan seuraavaan tai edelliseen kuvaan ilman, että joutuu aina välillä sulkemaan valitun kuvan.

Testaajille annettiin samat tehtävät kuin prototyypin käytettävyydestissä ja kaikki tehtävät onnistuivat hyvin kaikilta käyttäjiltä.

Testaajille annetut tehtävät:

- Lähetä tarjouspyyntö.
- Tarkastele yrityksen esimerkkikuvia ja suurennat kuva paremmin näkyväksi.

Valmiin sivuston think aloud -käytettävyydestissä ei ilmennyt ongelmia tai korjattavia asioita, joten sivusto oli käytettävyydestin jälkeen valmis luovutettavaksi asiakkaalle.

### **11.5.3 Sivuston visuaalinen suunnitelma**

Ennen ohjelmoinnin aloittamista prototyypin pohjalta kehitettiin sivustosta vielä Adobe Photoshop -ohjelmalla tarkka visuaalinen suunnitelma, joka on nähtävillä liitteessä 8. Suunnitelman tarkoituksena oli luoda mahdollisimman tarkka konkreettinen kuva siitä, miltä sivusto tulisi lopulta näyttämään.

Visuaalisessa suunnitelmassa prototyyppiä kehitettiin lisäämällä sivustolle väriä käyttämällä logossa olevaa punaista väriä ja eloa sivustolle tuotiin lisäämällä taustakuvia teks-

tien taustalle. Visuaalista suunnitelmaa pidettiin prototyypin ohella koko ajan esillä, kun sivustoa ohjelmoitiin, jotta lopputulos olisi mahdollisimman lähellä alkuperäistä suunnitelmaa.

#### 11.5.4 Sivuston ohjelmointi

Sivuston ohjelmointi toteutettiin Photoshopilla tehdyn suunnitelman pohjalta. Ensin lähdettiin tarkastelemaan, millaisia eri alueita sivuilla on. Suunnitelman perusteella sivu jakautui seuraaviin seitsemään alueeseen: header eli navigaatio, tämän jälkeen tuleva kuvakaruselli, Tietoa meistä -osio, Mikrosegmentointi-osio, kuvagalleria, tarjouspyyntö sekä footer. Sivuston suunnitelma pidettiin koko ajan esillä, koska sen oli tarkoitus toimia ohjenuorana sivuston ohjelmoinnille. Suunnitelmana ohjelmoinnin etenemiselle oli, että ensin luotaisiin sivuston seitsemän eri aluetta, jotta saataisiin jokainen osio omalle paikalleen ja tämän jälkeen pureuduttaisiin tyyleihin ja muihin yksityiskohtiin.

Sivuston rakennuksessa lähdettiin liikenteeseen perustekijöistä luomalla projektille oma kansio nimellä mcv ja tämän sisään sivupohja index.html. Tässä vaiheessa index.html-tiedostoon lisättiin sivuston toiminnan kannalta vain välttämättömimmät elementit. Sivuston kieli tulisi olemaan suomi, joten lisättiin se lang-attribuuttiin. Tämän lisäksi annettiin sivustolle kuvaava otsikko yrityksen nimen mukaan sekä välttämätön charset-attribuutti (kuva 14).

```
<!DOCTYPE html>
<html lang="fi">
  <head>
    <title>MCV remontointi</title>
    <meta charset="utf-8">
  </head>
  <body>
  </body>
</html>
```

Kuva 14. Index.html-tiedoston pohja

Myöhemmässä vaiheessa sivustolle tullaan luomaan useita eri tyyli-tiedostoja, joten tyy-leille oli järkevää luoda oma kansio nimellä css. Kansion sisään luotiin ensimmäinen tyyli-tiedosto style.css. Kyseinen tyyli-tiedosto liitettiin index.html-tiedoston head-osaan (kuva 15).

```
<link rel="stylesheet" href="css/style.css" type="text/css">
```

Kuva 15. Tyyli-tiedoston liittäminen

Tässä vaiheessa oli jo hyvä lisätä tyylitiedostoon muutamat tärkeät koko sivustoon vaikuttavat tyylit. Responsiivisuus on välttämätön piirre verkkosivuille, koska sivuja selataan yhä useammin puhelimilla tai tableteilla isojen näyttöjen sijaan. Sivuston sisällön asettelussa ja tätä kautta myös responsiivisuudessa päätettiin hyödyntää gridejä, joiden avulla sivu jaettiin kahteentoista pystysuoraan yhtä suureen osioon. Gridit helpottavat eri elementtien sijoittelua sivuille ja niiden avulla pystytään helposti kontrolloimaan sivujen käyttäytymistä erikokoisilla näytöillä.

Ensimmäiseksi lisättiin tyyli, joka varmistaa, että paddingit sekä borderit sisältyvät elementtien kokonaispituuteen sekä -leveyteen. Kyseinen tyyli, box-sizing: border-box, asetettiin kaikille HTML-elementeille (kuva 16). Tämän jälkeen määriteltiin tyylit, jotka jakavat sivun kahteentoista samankokoiseen kolumniin. Jako suoritettiin jakamalla sivun kokonaisleveys, 100 %, kahteentoista osaan. Tällöin yhden kolumnin leveys laskettiin kaavalla  $100 \% / 12 = 8,33 \%$  ja näin ollen kahden kolumnin leveys yhteensä saatiin johdettua edellisestä kaavalla  $2 \times 8,33 \% = 16,66 \%$  ja niin edelleen. Tyyleihin luotiin kaksitoista eri luokkaa, joilla jokaisella on eri leveys. Kyseisiä luokkia hyödyntämällä oli tarkoitus kontrolloida sivuja helposti kaikilla näyttöko'illa. Luokat nimettiin kuvaavasti numeroin, jotta niiden käyttö olisi mahdollisimman yksinkertaista (kuva 16).

Seuraavaksi lisättiin luokka column, joka mahdollistaa eri kokonaisuuksien sijoittumista sivuilla rinnakkain (kuva 16). Kyseistä luokkaa tullaan käyttämään jokaisen edellä luodun alueen leveyttä määrittävän luokan yhteydessä. Näin ollen esimerkiksi kaksi eri elementtiä, joilla molemmilla on luokka three (`class="three"`), sijoittuvat sivuilla rinnakkain eivätkä allekkain saadessaan myös luokan column (`class="three column"`).

Sivujen suunnitelman mukaisesti osa sisällöstä sijoittui sivun koko leveydelle, kun taas osa sijoittui kapeammalle alueelle sivun keskelle. Kyseinen kapeampi alue on saman levyinen sivuston jokaisella eri alueella, jossa sisältö ei ulotu sivun laidasta laitaan. Oli yksinkertaisinta luoda luokka wrap, joka määrittää sivun leveyden kapeammilla sisältöalueilla (kuva 16). Kyseistä luokkaa hyödyntäen oli tarkoitus helposti saada yhteneväinen linja sivuston ulkonäölle.

Viimeiseksi tyyleihin lisättiin vielä ominaisuudet, jotka pitävät huolta siitä, että jokaisen alueen (header, footer, section) jälkeinen alue alkaa aina ikään kuin uudelta tyhjältä riviltä. Joissain tapauksissa uusi alue pyrkii sijoittumaan samalle riville edellisen alueen kanssa (`float: left`), mutta koska uusi alue haluttiin aina uudelle riville edellisen alle, tarvittiin tyyli alueelle header:after ja muille samassa yhteydessä mainituille (kuva 16).

```

*, :after, :before {
  -moz-box-sizing: border-box;
  -webkit-box-sizing: border-box;
  box-sizing: border-box;
}

.one { width:8.33%!important; }
.two { width:16.66%!important; }
.three { width:25%!important; }
.four { width:33.3333%!important; }
.five { width:41.66%!important; }
.six { width:50%!important; }
.seven { width:58.33%!important; }
.eight{ width:66.66%!important; }
.nine{ width:75%!important; }
.ten{ width:83.33%!important; }
.eleven { width:91.66%!important; }
.twelve, .twelve { width:100%!important; }

.column {
  float:left;
}

.wrap {
  width: 1300px;
  margin: 0 auto;
  max-width: 100%;
}

header:after, section:after, footer:after {
  content: "";
  display: table;
  clear: both;
}

```

Kuva 16. Tärkeät koko sivuston toimintaan vaikuttavat tyylit

## Header

Sivuston rakennus aloitettiin sivun yläosasta, joten ensimmäiseksi luotiin header eli navigaatio (kuva 17). Sivuston suunnitelman mukaisesti navigaatio ei ulottunut reunasta reunaan, joten tässä kohtaa käytettiin edellä tyyleihin luotua wrap-luokkaa. Navigaatio asetettiin kokonaisuudessaan sen divin sisälle, jolle wrap-luokka annettiin. Navigaatioon sisällytettiin valmiiksi tarvittavat linkit sivuston muihin alueisiin. Href-attribuuteille asetettiin arvoiksi eri alueita kuvaavat id:t. Kyseiset id:t tulee muistaa lisätä jokaisen alueen yhteyteen, jotta linkitys saadaan myöhemmässä vaiheessa toimivaksi.

```

<header>
  <div class="wrap">
    <nav>
      <ul>
        <li><a href="#tieto">tietoa meistä</a></li>
        <li><a href="#mikro">mikrosementointi</a></li>
        <li><a href="#kuva">kuvagalleria</a></li>
        <li><a href="#tarjous">tarjouspyyntö</a></li>
        <li><a href="#yhteys">yhteystiedot</a></li>
      </ul>
    </nav>
  </div>
</header>

```

Kuva 17. Header-elementin luonti

## Kuvakaruselli

Seuraava alue sivuilla headerin jälkeen oli kuvakaruselli, joka asettui välittömästi navigaation alle. Suunnitelman mukaan kuvan oli tarkoitus olla vaihtuva, jotta siinä pystyttäisiin esittelemään useita kuvia heti käyttäjän saapuessa sivuille. Tässä kohtaa hyödynnettiin Bootstrapia ja sen carousel-liitännäistä, joka mahdollistaa näyttävän ja toimivan kuvien vaihtumisen eli kuvakarusellin varsin helposti. Karusellin liittäminen sivuille onnistui kätevimmin W3schools-verkkosivulta löytyvää helposti seurattavaa Bootstrap carousel -ohjeistusta hyödyntäen (W3schools.com 1999-2018f).

Jotta Bootstrap framework voitiin ottaa käyttöön, tuli projektiin liittää Bootstrapin omat tyyli- ja JavaScript-tiedostot sekä jQuery. Kyseiset tiedostot saa lataamalla Bootstrapin, mutta tässä projektissa hyödynnettiin toista tekniikkaa ja otettiin tarvittavat tiedostot käyttöön CDN:n kautta (Content Delivery Network). Tiedostot lisättiin index.html-tiedoston head-osaan (kuva 18). Lisäksi head-elementtiin lisättiin metatieto, joka on tärkeä Bootstrapin responsiivisuuden kannalta (kuva 19).

```

<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

```

Kuva 18. Bootstrapin vaatimien tiedostojen liittäminen index.html-tiedostoon

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">

```

Kuva 19. Metatieto Bootstrapin responsiivisuuteen liittyen



Kun Bootstrapin vaatimat tiedostot oli otettu käyttöön, luotiin karusellialue. Kopioitiin edellä mainitulta W3schools-sivulta koodi kohdan Add Captions to Slides alta. Koodista haluttiin tarkalleen ottaen kopioida aloittavan tagin `<div class="container">` ja sitä vastaavan lopettavan `</div>` sisältö. Kopioitu koodi liitettiin index.html-tiedostoon headerin perään. Kyseisestä valmiista koodista poistettiin alun h2-otsikko tarpeettomana.

Seuraavaksi karuselliin lisättiin omat kuvat, jotta varmistuttiin sen oikeasta toiminnasta. Luotiin index.html-tiedoston rinnalle img-kansio, johon tallennettiin karuselliin halutut kolme kuvaa. Asetettiin koodin img-tageille polut haluttuihin kuviin ja poistettiin samalla alt-attribuutin arvot, jotka voidaan myöhemmin asettaa vastaamaan lopullisia kuvia (kuva 20). Kopioidussa koodissa mukana tuli myös jokaiselle kuvalle oma otsikko- sekä tekstikenttä. Kyseiset kentät tulitaisiin päivittämään tarkoituksenmukaisiksi työn myöhemmässä vaiheessa.

```
<div class="carousel-inner">
  <div class="item active">
    
    <div class="container">
      <div class="carousel-caption">
        <h1>Otsikko</h1>
        <p>Teksti</p>
      </div>
    </div>
  </div>

  <div class="item">
    
    <div class="container">
      <div class="carousel-caption">
        <h1>Otsikko</h1>
        <p>Teksti</p>
      </div>
    </div>
  </div>

  <div class="item">
    
    <div class="container">
      <div class="carousel-caption">
        <h1>Otsikko</h1>
        <p>Teksti</p>
      </div>
    </div>
  </div>
</div>
```

Kuva 20. Carousel kuvat ja tekstit

Otsikko- ja tekstikenttien yhteyteen haluttiin myös luoda linkit, jotka ohjaisivat tiettyyn kohtaan sivuja. Linkki luotiin tekstikentän alle ja sille annettiin työn tässä vaiheessa ulkonäön vuoksi luokkia, jotka viittaavat Bootstrapin omiin tyyli-tiedostoihin (kuva 21). Kyseisiä tyyliä oli tarkoitus muokata myöhemmin sivuston kokonaisuuteen sopivaksi. Myös linkkien

href-attribuuteille oli tarkoitus antaa myöhemmin id:t, jonka perusteella ne voidaan kohdistaa haluttuihin kohtiin sivustolla.

```
<div class="item active">
  
  <div class="container">
    <div class="carousel-caption">
      <h1>Otsikko</h1>
      <p>Teksti</p>
      <p><a class="btn btn-lg btn-primary" href="#" role="button">
        Lue lisää</a></p>
    </div>
  </div>
</div>
```

Kuva 21. Linkkien lisäys karusellin tekstien yhteyteen

### Tietoa meistä

Suunnitelman mukaan Tietoa meistä -alueen sisältö ei myöskään sijoittunut laidasta laitaan, joten myös sille annettiin wrap-luokka. Tässä osiossa alettiin niin ikään hyödyntää luokkia, jotka määrittävät kolumnien leveyden. Koska alueelle haluttiin kaksi samankokoista osiota rinnakkain, annettiin molemmille osioille kaksi eri luokkaa: six ja column (class="six column"). Tekstialueiden alapuolelle oli tarkoitus tulla neljä kuvaa, joten tämä alue jaettiin vastaavasti neljään samankokoiseen alueeseen antamalla jokaiselle alueelle luokat three ja column (class="three column"). Column-luokan oli siis tarkoitus pitää huoli siitä, että esimerkiksi tässä yhteydessä kaikki kuvat asettuvat sivulla vieretysten. Ilman column-luokkaa kuvat olisivat asettuneet allekkain, joten kyseistä luokkaa tulisi käyttää sivustolla kaikissa niissä yhteyksissä, joissa halutaan sijoittaa eri elementtejä vierekkäin (kuva 22).

```

<section class="tietoa_meista" id="tieto">
  <div class="wrap">
    <h2>Tietoa meistä</h2>
    <div class="col-container">
      <div class="six column">
        <p>Olemme pääkaupunkiseudulla toimiva kylpyhuoneremontointiin
erikoistunut yritys. Meillä on yli kymmenen vuoden kokemus
erilaisista kylpyhuoneremonteista pitäen sisällään sekä saneeraukset
että uudiskohteet. Meillä on VTT-sertifikaatti sekä
paljon referenssejä. <br /><br />
<a href="#yhteys">Ota yhtettä</a> niin kerromme mielellämme lisää!</p>
      </div>
      <div class="six column">
        <ul>
          <li>Kylpyhuoneremontit</li>
          <li>Laatoitustyöt</li>
          <li>Mikrosegmentointi</li>
          <li>yms.</li>
        </ul>
      </div>
    </div>
    <div class="three column">
      
    </div>
    <div class="three column">
      
    </div>
    <div class="three column">
      
    </div>
    <div class="three column">
      
    </div>
  </div>
</section>

```

Kuva 22. Tietoa meistä -alue

Tietoa meistä -alueelle annettiin myös `id="tieto"`, jotta navigaation linkitys alueelle toimisi. Vastaavat yksilölliset `id:t` tulee antaa myöhemmässä vaiheessa myös kaikille muille alueille sen mukaan, kuinka ne on nimetty navigaation `href`-attribuutissa. Alueen neljä uutta kuvaa leikattiin kokoon 250 x 250 px ja tallennettiin sivuston `img`-kansioon. Jotta kuvat pysyisivät aina samalla rivillä riippumatta ylläolevan tekstin määrästä, tarvittiin tässä vaiheessa myös muutama uusi tyyli (kuva 23). Kyseinen luokka lisättiin alueen tekstiosioden ympärille ja sillä haluttiin pitää huolta siitä, että tekstiosiot venyvät samankorkuisiksi toisiinsa nähden, vaikka ne sisällöltään olisivat erikorkuiset. Tämän johdosta myös kuvat pysyvät aina rinnakkain samassa linjassa.

```

section.tietoa_meista .col-container {
  display: flex;
  width: 100%;
}

```

Kuva 23. Rinnakkaisten tekstialueiden venytys samankorkuisiksi

## Mikrosegmentointi

Mikrosegmentointi-alue oli suunnitelman mukaan jaettu kahteen samankokoiseen osaan, joten molemmille annettiin kaksi eri luokkaa: six ja column (class="six column") (kuva 24). Asiakas halusi sivuille linkin heidän käyttämäänsä mikrosegmenttiin, joten lisättiin se, ja oikealle vastaavasti tekstiä. Myös tämän alueen sisältö on taustakuvaa lukuun ottamatta kapeammalla alueella, joten alueen ympärille lisättiin jälleen wrap-luokka.

```
<section class="mikrosegmentointi" id="mikro">
  <div class="wrap">
    <h2>Mikrosegmentointi</h2>
    <div class="six column">
      <a href="https://dekotuote.fi/novacolor-italia-tuotteet/" target="_blank">Tutustu lisää</a>
    </div>
    <div class="six column">
      <p>Kylpyhuoneiden mikrosegmentointi on erityisalaamme. Tekniikka mahdollistaa modernin ilmeen kylpyhuoneen pintoihin varsin kustannustehokkaasti. Voit tutustua mikrosegmentoinnilla saatavaan ilmeeseen kuvagalleriamme ensimmäisessä kuvassa, jossa kylpyhuoneen pinnat on käsitelty mikrosegmentillä.</p>
    </div>
  </div>
</section>
```

Kuva 24. Mikrosegmentointi-alueen perusta

## Kuvagalleria

Kuvagallerian luonnissa hyödynnettiin Lightboxin tarjoamaa valmista pohjaa, joka löytyy osoitteesta <http://lokeshdhakar.com/projects/lightbox2/>. Ohjelmoinnissa edettiin sivun ohjeiden mukaisesti eli ensimmäiseksi ladattiin Lightbox ja purettiin se omalle koneelle. Tämän jälkeen kopioitiin tarvittavat css- sekä JavaScript -tiedostot ja liitettiin ne omaan projektiin css- ja js-kansioihin. Samanaikaisesti tehtiin linkit kyseisiin tiedostoihin. Linkki tyylitiedostoon index.html:n head-osaan ja linkki JavaScript-tiedostoon vastaavasti ennen bodyn loppua (kuvat 25, 26).

```
<link rel="stylesheet" href="css/lightbox.css" type="text/css">
```

Kuva 25. Linkki Lightboxin tyylitiedostoon

```
<script src="js/lightbox.js"></script>
</body>
```

Kuva 26. Linkki Lightboxin JavaScript-tiedostoon

Kuvagalleria asettuu myös kapeammalle alueelle, joten lisättiin wrap-luokka. Gallerian kuvia on suunnitelman mukaan tarkoitus tulla neljä rinnakkain. Näin ollen jokainen yksittäinen kuva asettuu luokkien three ja column sisään (class="three column"). Toimiakseen tarkoituksenmukaisesti tarvitsee Lightbox jokaisesta kuvasta kaksi eri kokoa: pienen ja suuren. Pieni kuva näkyy sivuilla oletuksena, ja suuri kuva avautuu pientä klikkaamalla. Asiakkaan antamista referenssikuvista muokattiin näin ollen pienet sekä suuret versiot,

jotka tallennettiin projektin img-kansioon. Kaikki kuvat leikattiin toisiaan vastaaviin pieneen ja suureen kokoon, jotta niiden ulkonäkö sekä käyttäytyminen sivuilla olisi mahdollisimman yhteneväistä.

Jokainen pieni kuva lisättiin img-elementtinä, josta on linkki suureen kuvaan (kuva 27). Jokainen kuvapari sijoitettiin oman divin sisään, jotta kuvat saatiin sijoiteltua halutusti rinnakkain. Suurien kuvien yhteyteen oli tärkeää lisätä data-lightbox-attribuutti, jonka arvo on sama jokaisella kuvalla (gallery). Kyseinen attribuutti pitää huolen siitä, että sivusto näkee kuvat kokonaisuutena, jolloin jokaisesta avautuvasta suuresta kuvasta on mahdollisuus selata kuvia eteen- ja/tai taaksepäin. Gallerian oikeaa toimintaa kokeiltiin alkuun muutamalla kuvalla (kuva 28).

```
<div class="three column">
  <a href="img/remontti-suuri3.jpg" data-lightbox="gallery">
    </a>
  </div>
```

Kuva 27. Gallerian kuvien lisäys

```
<section class="galleria" id="kuva">
  <div class="wrap">
    <h2>Kuvagalleria</h2>
    <div>
      <div class="three column">
        <a href="img/remontti-suuri3.jpg" data-lightbox="gallery">
          </a>
        </div>
      <div class="three column">
        <a href="img/remontti-suuri1.jpg" data-lightbox="gallery">
          </a>
        </div>
      <div class="three column">
        <a href="img/remontti-suuri4.jpg" data-lightbox="gallery">
          </a>
        </div>
      <div class="three column">
        <a href="img/remontti-suuri12.jpg" data-lightbox="gallery">
          </a>
        </div>
    </div>
  </div>
</section>
```

Kuva 28. Galleria-alue neljällä kuvalla

## Lomake

Suunnitelman mukaan tämä alue rakentui siten, että lomake vie vasemman puoliskon sivusta ja oikealle jätetään yhtä suuri tila lomakkeeseen liittyvälle ohjeistukselle. Tuttuun tapaan luotiin wrap-luokan omaava alue, jonka sisään lisättiin kaksi samankokoista vierek-



käistä aluetta six- ja column-luokkien avulla (class="six column") (kuva 29). Tässä vaiheessa luotiin myös uusi luokka, flex-container (class="flex-container"), johon lisättiin sisällön sijoittelun kannalta tärkeitä tyylejä (kuva 30).

```
<section class="tarjouspyynto" id="tarjous">
  <div class="wrap">
    <h2>Tarjouspyyntö / Yhteydenotto</h2>
    <div class="flex-container">
      <div class="six column">
        <form>
          <div class="form-group">
            <label for="nimi">nimi:</label>
            <input type="text" class="form-control" id="nimi" name="nimi">
          </div>
          <div class="form-group">
            <label for="osoite">osoite:</label>
            <input type="text" class="form-control" id="osoite" name="osoite">
          </div>
          <div class="form-group">
            <label for="email">sähköposti:</label>
            <input type="text" class="form-control" id="email" name="email">
          </div>
          <div class="form-group">
            <label for="puhelin">puhelinnumero:</label>
            <input type="text" class="form-control" id="puhelin" name="puhelin">
          </div>
          <div class="form-group">
            <label for="viesti">viesti:</label>
            <textarea class="form-control" rows="5" id="viesti" name="viesti">
            </textarea>
          </div>
          <br />
          <button type="submit" class="btn btn-default">LÄHETÄ</button>
        </form>
      </div>
      <div class="six column">
        <p>Tähän tulee tekstiä</p>
      </div>
    </div>
  </div>
</section>
```

Kuva 29. Lomake-alue

```
.flex-container {
  display: flex;
  flex-wrap: wrap;
}
```

Kuva 30. Lomake-alueen tarpeelliset tyylit

Koska Bootstrap on jo kuvakarusellin yhteydessä lisätty projektiin, hyödynnettiin sen ominaisuuksia myös lomakkeen yhteydessä. Sivulta [https://www.w3schools.com/bootstrap/bootstrap\\_forms.asp](https://www.w3schools.com/bootstrap/bootstrap_forms.asp) löytyy Vertical form -malli, jonka mukaan sivuille tuleva lomake rakennettiin. Esimerkin mukaisesti luotiin label-elementti ennen jokaista input-elementtiä. Tämän jälkeen asetettiin molemmat edellä mainitut elementit form-group-luokan sisälle (class="form-group"). Labelin for-attribuutin arvon tulee olla sama kuin input-elementin id-attribuutin, jotta käyttäjä saa paremman käyttökokeimuksen. Jokaiselle input-elementille sekä textarealle lisättiin vielä form-control-luokka

(class="form-control"), jotta saatiin lomake näyttämään hyvältä. Lomakkeella käytettävät luokat ovat Bootstrapin valmiita tyylejä, joita on kuitenkin mahdollisuus muokata halutesaan. Lomakkeen toiminnollisuuteen ei keskitytty vielä tässä vaiheessa, vaan se tullaan tekemään myöhemmin PHP-ohjelmoinnin avulla.

## Footer

Footer jakautuu edellisen lomake-osion tavoin kahteen yhtä leveään alueeseen, joiden alla on vielä yksi rivi keskitettyä tekstiä. Sisällöt asetettiin jälleen wrap-luokan (class="wrap") sisään ja luotiin halutut alueet six column- ja twelve column -luokkien avulla (kuva 31). Vasemmanpuoleisen tekstiosion ensimmäisen rivin oli tarkoitus olla katseenkiinnittäjä, joten lisättiin sen ympärille valmiiksi span-elementti, jolla pystyttäisiin myöhemmin kohdistamaan muusta tekstistä poikkeavia tyylejä kyseiselle alueelle.

```
<footer class="foot" id="yhteys">
  <div class="wrap">
    <div class="six column">
      <p><span>MCV Remontointi</span><br>tekstiä</p>
    </div>
    <div class="six column">
      Puhelin: 040 123 4567
      Sähköposti: esimerkki@gmail.com
    </div>
    <div class="twelve column">
      <p>&copy; 2018 MCV Remontointi</p>
    </div>
  </div>
</footer>
```

Kuva 31. Footer-alueen sisällön jakautuminen

Oikealle puolelle haluttiin yhteystiedot ja niitä korostamaan päätettiin lisätä kuvakkeet. Kuvakkeet lisättiin sivuille hyödyntämällä Fontellon ikoneja (Fontello). Fontellon sivuilta valittiin halutut ikonit, ladattiin ne ja tallennettiin omalle koneelle. Tässä vaiheessa oli jo tiedossa, että sivuilla tullaan tarvitsemaan myös jonkinlaista menu-ikonia, joka lisätään headeriin pienillä näytöillä, jotta navigaatio saadaan näkyviin. Samassa yhteydessä siis valittiin ikoni myös kyseistä tarkoitusta varten, jotta saatiin kaikki tarvittavat Fontellon ikonit yhdellä latauskerralla. Omalle koneelle tallennuksen jälkeen tiedoston sisällöstä kopioitiin koko font-kansio ja liitettiin se oman projektin juureen. Tämän lisäksi css-kansiosta löytyvä fontello.css kopioitiin ja liitettiin oman projektin css-kansion alle. Linkki Fontellon tyylitiedostoon lisättiin index.html:n head-osioon Lightbox-tyylitiedostolinkin ylle (kuva 32).

```
<link rel="stylesheet" href="css/fontello.css" type="text/css">
```

Kuva 32. Linkki Fontellon tyylitiedostoon

Ikonit liitettiin sivuille Fontellon valmiiksi luomien luokkien avulla. Luokkien nimet selviävät omaan projektiin liitetyn fontello.css-tiedoston lopusta (kuva 33). Kuvan ylin luokka viittaa navigaatioon tarkoitettuun ikoniin, jota ei vielä tässä kohtaa tarvittu. Jotta ikonit saadaan näkyviksi, täytyy kyseiset luokat liittää sivuille niihin kohtiin, joihin ikonin haluaa sijoittaa. Näin ollen luokat lisättiin kukin omalle rivilleen siihen liittyvän tekstin yhteyteen (kuva 34). Itse teksti sijoitettiin valmiiksi span-luokan sisään, jotta sille pystytään myöhemmin antamaan erilaisia tyylejä vaikuttamatta itse ikoniin.

```
.icon-menu:before { content: '\e800'; }  
.icon-phone:before { content: '\e825'; }  
.icon-mail-alt:before { content: '\f0e0'; }
```

Kuva 33. Ladattujen Fontello-ikonien luokat

```
<footer class="foot" id="yhteys">  
  <div class="wrap">  
    <div class="six column">  
      <p><span>MCV Remontointi</span><br>tekstiä</p>  
    </div>  
    <div class="six column">  
      <div class="icon-phone"><span>Puhelin: 040 123 4567</span></div>  
      <div class="icon-mail-alt"><span>Sähköposti: esimerkki@email.com</span></div>  
    </div>  
    <div class="twelve column">  
      <p>&copy; 2018 MCV Remontointi</p>  
    </div>  
  </div>  
</footer>
```

Kuva 34. Footerin ikonien lisäys

### Tyylien lisääminen

Kun sivuston perusrakenteet ja sisältö saatiin kuntoon, siirryttiin muokkaamaan sivuja yksityiskohtaisempien tyylien avulla. Sivustolle annettiin taustaväriksi vitivalkoisen sijaan aavistuksen harmaaseen taittuva sävy, ja footeriin haettiin sopiva punainen, jotta saatiin kyseinen alue erottumaan sivuston muusta osasta. Suunnitelman mukaan sivuille oli tarkoitus tulla vaaleaa taustaa sekä tummempaa taustakuvaa alueesta toiseen vaihdellen, jotta eri aihesisällöt saataisiin erottumaan toisistaan paremmin. Taustakuvat muutamalle alueelle lisättiin tässä vaiheessa, jotta saatiin selkeyttä ja rakennetta sivuille. Mikrosemontointi- sekä lomake-alueiden taustakuvat lisättiin tyylitiedoston puolelta ja kuville tehtiin tummentavat filterit, jotta tekstit erottuvat entistä paremmin (kuva 35). Samalla sivuston kaikille osioille annettiin tilaa sekä ylä- että alapuolelle määrittelemällä paddingit.



```

section.tarjouspyynto {
  background-image: url('../img/remontti-taustakuva.jpg');
  background-position: 50% 50%;
  background-size: cover;
  position: relative;
  padding: 100px 0;
}

section.tarjouspyynto div {
  position: relative;
  z-index: 2;
}

section.tarjouspyynto:before {
  content: "";
  position: absolute;
  top: 0;
  bottom: 0;
  left: 0;
  right: 0;
  background: rgba(13,13,13, 0.5);
  z-index: 1;
}

```

Kuva 35. Taustakuva, ylä- ja alapaddingit sekä taustakuvan tummennus

Seuraavaksi siirryttiin muokkaamaan headeria. Tässä yhteydessä nollattiin monia tarpeettomia oletustyylejä ja annettiin tilalle muita kokonaisuuteen paremmin sopivia tyylejä (kuva 36). Navigaatiopalkista tehtiin musta valkoisella tekstin värillä, ja navigaation sisältävät lisäelementit asetettiin rinnakkain palkin keskelle. Myös linkkien tyylejä muokattiin, jotta ne olisivat värimaailmaltaan sivuihin sopivat, ja yrityksen logo lisättiin vasempaan yläkulmaan.

```

header {
  position: fixed;
  z-index: 20;
  background-color: #000;
  opacity: ;
  left: 0;
  right: 0;
  top: 0;
}

header nav ul {
  margin: 20px 10px;
  padding: 0;
  list-style-type: none;
  display: flex;
  justify-content: space-between;
}

nav ul li:hover a {
  color: #b30000;
  text-decoration: none;
}

header nav ul li a {
  text-transform: uppercase;
  line-height: 70px;
  text-decoration: none;
  color: #fff;
  padding: 0 20px;
  display: block;
}

```

Kuva 36. Headerin tyylejä

Taustakuvien ollessa tummat jäi musta teksti niiden päällä lähes näkymättömiin. Kaikille taustakuvien päällä sijaitseville teksteille määriteltiin valkoinen väri. Tämän lisäksi otsikoita korostettiin varjostuksella sekä suurilla kirjaimilla, jotka toteutettiin kahden eri luokan avulla (kuva 37). Kyseisten luokkien avulla määriteltiin tyylit tyylitiedoston puolella (kuva 38). Kuvassa näkyvät tyylit voisi yhdistää myös yhden ja saman luokan alle. Subheader-luokkaa kuitenkin hyödynnettiin myös kaikissa muissa sivuston otsikoissa, jotka eivät kaikki ole väriltään valkoisia, minkä vuoksi tässä tapauksessa luokkia muodostui kaksi.

```

<div class="subheader white-heading">
  <h2>Mikrosementointi</h2>
</div>

```

Kuva 37. Otsikon luokat

```

section.mikrosementointi .subheader {
  padding: 0 0 50px 0;
  text-align: center;
  text-transform: uppercase;
}

section.mikrosementointi .white-heading {
  color: #fff;
  text-shadow: 2px 2px 2px #000;
}

```

Kuva 38. Otsikko-luokkien tyylit

Tässä yhteydessä haettiin myös sopivat fontit Google Fonts -verkkosivuilta <https://fonts.google.com/>. Sivulla tullaan käyttämään kolmea eri fonttia. Linkki valittuihin fontteihin lisättiin Google Fonts -sivujen ohjeiden mukaan index.html:n head-osaan (kuva 39). Kyseinen linkki sijoitettiin jo aiemmin lisättyjen tyylitiedostolinkkien ylle. Tyylitiedoston puolella vastaavasti määriteltiin, missä yhteydessä mitäkin fonttia käytetään. Sivulle valittiin yksi koko sivua koskeva fontti, jonka lisäksi h2-otsikoille määriteltiin oma fontti korostusta varten (kuva 40).

### 3 Families Selected

Your Selection Clear All ↗ ⬇

Open Sans Condensed ⊖
Open Sans ⊖
Archivo Black ⊖

EMBED
CUSTOMIZE
Load Time: Moderate

#### Embed Font

To embed your selected fonts into a webpage, copy this code into the <head> of your HTML document.

STANDARD
@IMPORT

```

<link href="https://fonts.googleapis.com/css?family=Archivo+Black|Open+Sans|Open+Sans+Condensed:300" rel="stylesheet">

```

#### Specify in CSS

Use the following CSS rules to specify these families:

```

font-family: 'Archivo Black', sans-serif;
font-family: 'Open Sans', sans-serif;
font-family: 'Open Sans Condensed', sans-serif;

```

Kuva 39. Ohjeet Google Fontsin käyttöönottoon

```
body {
  font-family: 'Open Sans', sans-serif;
}

h2 {
  font-family: 'Archivo Black', sans-serif;
}
```

Kuva 40. Fonttien käytön määrittely

Seuraavaksi siirryttiin muokkaamaan kuvakarusellin aluetta halutuksi. Ensimmäiseksi karuselli muokattiin ulottumaan koko näytön leveydelle, jotta yleisilmeestä saatiin mahdollisimman näyttävä (kuva 41). Muilta osin keskityttiin muokkaamaan tekstien ja linkkien ko-koa, värimaailmaa ja sijoittumista. Esimerkiksi linkistä tehtiin sivujen värimaailmaan sopiva punainen läpikuultavalla ominaisuudella (kuva 42). Myös tekstialueita päästiin muokkaamaan samaan tapaan koodissa jo valmiiksi esiintyvien luokkien avulla. Kyseiset luokat viittaavat Bootstrapin omiin tyylietodostoihin, joten on tärkeää, että linkki itse luotuun tyylietodostoon (tässä projektissa style.css) tulee index.html:n head-osassa Bootstrapin tyylietodostolinkin jälkeen. Näin omilla tyyleillä saadaan ylikirjoitettua Bootstrapin tyyliä.

```
.container {
  width: 100%;
  padding: 0px;
}
```

Kuva 41. Kuvakaruselli-alueen leveys

```
.carousel-inner .carousel-caption a.btn {
  background: rgba(255, 1, 1, 0.6);
  border: none;
}
```

Kuva 42. Läpikuultava linkki

Mikrosementointi-osion molemmat puoleiset tekstit haluttiin keskittää samaan linjaan toisiinsa nähden ja pystysuunnassa keskelle taustakuvaa. Tätä varten tekstialueiden ympärille luotiin muutama eri luokka, joille asetettiin tarvittavat tyyliä haluttua lopputulosta silmällä pitäen (kuva 43). Luokkien tyyliä määriteltiin tuttuun tapaan tyylietodoston puolella (kuva 44).

```

<div class="flex">
  <div class="six column relative">
    <div class="text-box absolute">
      <a href="https://dekotuote.fi/novacolor-italia-tuotteet/" target="_blank">
        Tutustu lisää</a>
      </div>
    </div>
    <div class="six column text dark relative">
      <div class="absolute">
        <p>Tekstiä</p>
      </div>
    </div>
  </div>
</div>

```

Kuva 43. Mikrosegmentointi-osion tekstialueiden asetteluun vaadittavat luokat

```

section.mikrosegmentointi .relative {
  position: relative;
}

section.mikrosegmentointi .absolute {
  margin: auto;
  position: absolute;
  top: 50%;
  transform: translateY(-50%);
  left: 0;
  right: 0;
}

section.mikrosegmentointi .flex {
  display: flex;
  align-items: center;
  padding-bottom: 50px;
  padding-top: 50px;
}

section.mikrosegmentointi a {
  display: block;
  text-align: center;
  padding-top: 40px;
}

```

Kuva 44. Mikrosegmentointi-osion tekstialueiden asettelun tyylit

Kuvagalleria, lomake ja footer olivat jo hyvällä mallilla tyylien osalta, joten niihin ei tässä vaiheessa tarvinnut juurikaan tehdä lisäyksiä. Kun sivuston yleisilme oli haluttu, oli seuraava askel keskittyä responsiivisuuteen ja sen vaatimiin tyylimuutoksiin.

## Responsiivisuus

Sivuston responsiivisuutta ja sitä varten tarvittavia muutoksia lähdettiin selvittämään selaimen kautta, jossa sivusto oli avoinna. Selaimen ikkunan kokoa lähdettiin vähitellen pienentämään ja tarkasteltiin, mitä sivujen sisällölle tapahtuu eri ko'issa. Responsiivisuutta varten css-kansioon luotiin kokonaan uusi tyylitiedosto mobile.css. Linkki tyylitiedostoon asetettiin index.html-tiedoston head-osaan kaikista alimmaiseksi tyylitiedostoksi. Kyseiseen tiedostoon oli tarkoitus tehdä sivuston vaatimia tyylillisiä muutoksia ikkunan koon pienentyessä eli siirryttäessä pienemmille näytöille, kuten tableteille tai puhelimille.

Mobile.css-tiedoston on tarkoitus sisältää media query -sääntöjä, jotka määrittelevät näytön koon ja sen, mitä tietyille elementeille tapahtuu kyseisessä koossa. Ensimmäisenä muutosta vaativana kohtana näyttöä pienennettäessä tuli vastaan kuvagalleria-alue. Gallerian kuvat ovat melko kookkaita ja ne alkoivat mennä toistensa päälle näytön pienentyessä. Ratkaisuna tälle luotiin ensimmäinen media query niillä näytön mitoilla, joissa gallerian kuvat vielä asettuivat hyvin toisiinsa nähden. Tarkoitus oli siis tehdä tyylimuutos alueelle ennen kuin niin sanottuja ongelmia alkoi ilmetä. Tyylimuutos toteutettiin siten, että neljän rinnakkaisen kuvan sijaan kuvat muuttuivatkin kolmeksi rinnakkaiseksi, jolloin niiden ympärille jäi taas enemmän tilaa, eivätkä ne menneet toistensa päälle (kuva 45). Jos kuvassa näkyvää three-luokkaa tarkastellaan style.css:n puolella, huomataan, että kyseisen luokan leveys on 25 %, jolloin kuvia mahtuu rinnakkain neljä. Tässä tehdyllä muutoksella siis muutetaan kyseisen luokan leveyttä suuremmaksi, jolloin kuvia mahtuu vierekkäin enää kolme.

```
@media only screen and (max-width: 1330px) {  
  section.galleria .three {  
    width: 33.3333% !important;  
    text-align: center;  
  }  
}
```

Kuva 45. Mobile.css-tiedoston ensimmäinen tarvittava media query

Mobile.css-tiedostossa käytetään siis samoja luokkia, joiden mukaan sivustolle on aiemmin määritetty tyyliä style.css-tiedostossa. Koska linkki mobile.css-tiedostoon asettuu style.css-tiedoston jälkeen, se ylikirjoittaa aiemmat tyylit ja määrittelee uudet näytön koon pienentyessä. Sivuston jokaisen alueen käyttäytyminen tarkasteltiin samaan tapaan näyttöä pikku hiljaa pienentäen. Aina kun uusi muutosta vaativa alue ilmeni, luotiin kyseiseen näytön kokoon viittaava uusi media query, jonka sisään kirjoitettiin tarvittavat uudet tyylit.

Melko pian näyttöä pienennettäessä vastaan tuli myös se, että navigaation linkit eivät enää mahtuneet rinnakkain yläpalkkiin. Ratkaisuna tähän alkuperäinen navigaatio piilotettiin näytön koon pienentyessä ja yläpalkkiin asetettiin kuvake, jonka valitsemalla navigaation saisi avattua allekkaisina linkkeinä.

Navigaation muokkaaminen aloitettiin luomalla app.js-tiedosto, joka tallennettiin js-kansioon. Tämän jälkeen index.html:n headerin sisään luotiin uusi div, joka sai luokakseen mobile-menu (kuva 46). Seuraavaksi style.css:n puolella määriteltiin tyyli kyseiselle luokalle (kuva 47). Tyylien avulla luotiin neliön muotoinen alue, joka on näkymättömissä ennen



kuin mobile.css:n puolella toisin määritellään. Siinä näytön koossa, jossa alkuperäinen navigaatio halutaan vaihtaa mobiilinäkymään, kirjoitetaan taas uusi media query ja sen sisään mobile-menu-luokan tyyliksi display: block, jotta luokalle asetetut tyylit tulevat näkyviin. Samassa media queryssa header nav saa tyylikseen display: none, koska alkuperäinen navigaatio halutaan vastaavasti piilottaa.

```
<header>
  <div class="wrap">
    <a class="navbar-brand" href="#" class="btn btn-link" role="button">
      </a>
      <div class="mobile-menu"></div>
    <nav>
```

Kuva 46. Mobile-menu-luokka

```
.mobile-menu {
  display: none;
  height: 70px;
  width: 70px;
  background-color: ;
  float: right;
}
```

Kuva 47. Mobile-menu-luokan tyylit

Siinä vaiheessa, kun Fontellosta ladattiin kuvakkeet footeriin, ladattiin valmiiksi myös kuvake mobiilinavigaatiota varten. Menu-kuvakkeen luokka käytiin tarkistamassa css-kansiosta löytyvän fontello.css-tiedoston lopusta. Halutun kuvakkeen luokka on icon-menu ja se lisättiin edelle luodun divin sisään span-elementtinä (kuva 48).

```
<div class="mobile-menu"><span class="icon-menu"></span></div>
```

Kuva 48. Menu-kuvakkeen lisäys

Navigaation toiminta oli tarkoitus toteuttaa JavaScriptillä, jossa hyödynnettäisiin jQuerya. Projektissa oli jo aiemmin otettu käyttöön jQuery, joten sitä ei tässä vaiheessa tarvinnut ladata uudelleen. Ennen app.js-tiedoston täydennystä lisättiin kuitenkin vielä muutama tarpeellinen ominaisuus. Index.html:n bodyn avaukseen lisättiin luokka nav-open. Tämän jälkeen muutamalle kyseisen luokan alla olevalle elementille käytiin lisäämässä tyyliä mobile.css:n puolelle samaan kohtaan, jossa mobiilinavigaatio tulee käyttöön (kuva 49).

```

.nav-open header nav {
  display: block;
  padding-top: 70px;
}
.nav-open header nav ul {
  flex-direction: column;
}

```

Kuva 49. Bodylle asetetun luokan tyylit

Kun tarvittavat luokat tyyleineen oli lisätty, oli aika lisätä sisältö app.js-tiedostoon. Tiedostoon kirjoitettavan koodin oli tarkoitus toimia siten, että näpäyttämällä aiemmin luotua mobiilipainiketta bodylle joko lisätään tai siltä poistetaan nav-open-luokka. Toisin sanoen JavaScriptillä toteutetaan navigaation aukeaminen ja sulkeutuminen vuoron perään saman napin painalluksella (kuva 50).

```

$(document).ready(function(){
  $('#header .mobile-menu').click(function(){
    $('#body').toggleClass('nav-open');
  });
});

```

Kuva 50. App.js-tiedoston sisältö

### Lomakkeen käsittely

Kuten aiemmin lomakkeen luonnin yhteydessä mainittiin, tullaan lomakkeen käsittely ja lähetys toteuttamaan PHP:llä. Ensimmäiseksi lomakkeen yhteyteen lisättiin tarvittavat attribuutit, jotka ohjaavat lomakkeen tiedot eteenpäin php-tiedoston käsiteltäviksi (kuva 51). Method-attribuutin post-arvo määrittää, että lomakkeelle kirjoitetut tiedot eivät tule näkyviin osoiteriville lähetyksen yhteydessä. Action-attribuutti vastaavasti määrittää sen tiedoston nimen ja sijainnin, jonne tiedot lähetetään. Php-koodia varten luotiin siis tiedosto nimeltä form.php, joka tallennettiin uuteen include-kansioon.

```

<form name="contactform" method="post" action="include/form.php">

```

Kuva 51. Lomakkeen lähetykseen tarvittavat attribuutit

Php-tiedoston puolella lähdettiin käsittelemään lomakkeelta tulleita tietoja. Tietojen käsittely aloitettiin tutkimalla sitä, onko käyttäjä syöttänyt sähköpostiosoitteensa. Se on yhteydenpidon kannalta tärkeä tieto, jotta asiakkaan viestiin pystytään vastaamaan. Joten jos asiakas on jättänyt sähköposti-kentän tyhjäksi, ohjataan hänet välittömästi takaisin täydentämään kyseinen tieto. Koodissa email viittaa lomakkeen email-kenttään ja sen sisältämään käyttäjän syöttämään tietoon. Koko lomakkeen käsittely tapahtuu kuvassa 52 näkyvien aaltosulkujen välissä. Mikäli käyttäjä on täydentänyt sähköpostiosoitteensa, jatketaan lomakkeen käsittelyä muilta osin.



```
<?php
if(isset($_POST['email'])) {
}
?>
```

Kuva 52. Tarkastetaan, onko lomakkeen sähköpostikenttä täydennetty

Seuraavaksi määriteltiin kaksi eri muuttujaa. Ensimmäinen näistä sisälsi sähköpostiosoitteen, johon lomakkeen tiedot haluttiin välittää. Toisen muuttujan arvoksi asetettiin otsikko, joka haluttiin näyttää jokaisen lomakkeen välityksellä tulleen sähköpostin yhteydessä (kuva 53). Tässä kohtaa määriteltiin myös error-viestien runko, joka tullaan näyttämään käyttäjälle, mikäli lomake sisältää virheitä, kuten tyhjäksi jätetyn sähköpostikentän.

```
$email_to = "m[redacted]n@live.com";
$email_subject = "Yhteydenotto mcv:n verkkosivuilta";
```

Kuva 53. Sähköposti- ja otsikko-muuttujat

Seuraava vaihe lomakkeen käsittelyssä oli hakea käyttäjän lomakkeelle syöttämät tiedot ja tallentaa ne eri muuttujiin. Kun lomakkeelta tulleet tiedot on tallennettu, ne täytyy validoida eli tarkastaa, ovatko ne oikeassa muodossa. Sääntöihin määriteltiin esimerkiksi, että nimi saa sisältää vain pieniä ja suuria kirjaimia ja sähköpostin tulee sisältää siihen vakiona kuuluvia merkkejä (kuva 54). Mikäli käyttäjän syöttämät tiedot eivät vastaa määriteltyjä sääntöjä, palautetaan yksityiskohtainen virheviesti, jonka perusteella käyttäjä pystyy tekemään tarvittavat korjaukset.

```
$name = $_POST['nimi'];
$address = $_POST['osoite'];
$email_from = $_POST['email'];
$phone = $_POST['puhelin'];
$message = $_POST['viesti'];

$error_message = "";
$email_exp = '/^[A-Za-z0-9._%]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}$/';

if(!preg_match($email_exp,$email_from)) {
    $error_message .= 'Syöttämäsi sähköpostiosoitte ei ole oikeassa muodossa.<br />';
}
```

Kuva 54. Lomakkeen tietojen tallennus ja käsittely

Kun käyttäjä on täydentänyt lomakkeen jokaisen kentän valideilla tiedoilla, lisätään tiedot sähköpostin sisältöön. Jotta sähköpostin sisältöä olisi helpompi tarkastella, jäsenneltiin eri tietoja otsikoin (kuva 55). Lopuksi tiedot lähetetään mail-funktiota hyödyntäen. Funktio sai neljä parametria: sähköposti, johon lomakkeen tiedot välitetään, otsikko, sisältö sekä headers-muuttujan arvot (kuva 56).

```

$email_message = "VIESTIN TIEDOT:\n\n";
$email_message .= "NIMI: ".clean_string($name)."\n";
$email_message .= "OSOITE: ".clean_string($address)."\n";
$email_message .= "SÄHKÖPOSTI: ".clean_string($email_from)."\n";
$email_message .= "PUHELIN: ".clean_string($phone)."\n";
$email_message .= "VIESTI: ".clean_string($message)."\n";

```

Kuva 55. Tietojen jäsentely sähköpostia varten

```

$headers = 'Lähtäjän sähköpostiosoite: '.$email_from."\r\n".
'Lähetä vastaus osoitteeseen: '.$email_from."\r\n".
'X-Mailer: PHP/' . phpversion();
@mail($email_to, $email_subject, $email_message, $headers);
?>

```

Kuva 56. Mail-funktion parametrit

Käyttäjän lomakkeelle täyttämät tiedot välittyvät siis samaan sähköpostiosoitteeseen, joka on mainittu verkkosivujen footerissa yhteydenottoa varten. Lomakkeen olemassaolo on kuitenkin tärkeää, koska se saattaa helpottaa mahdollisen asiakkaan yhteydenottoa. Kuitenkin asiakas voi itse valita, käyttääkö yhteydenottoon lomaketta vai lähestyykö suoraan sähköpostin välityksellä.

## 11.6 Tuotos

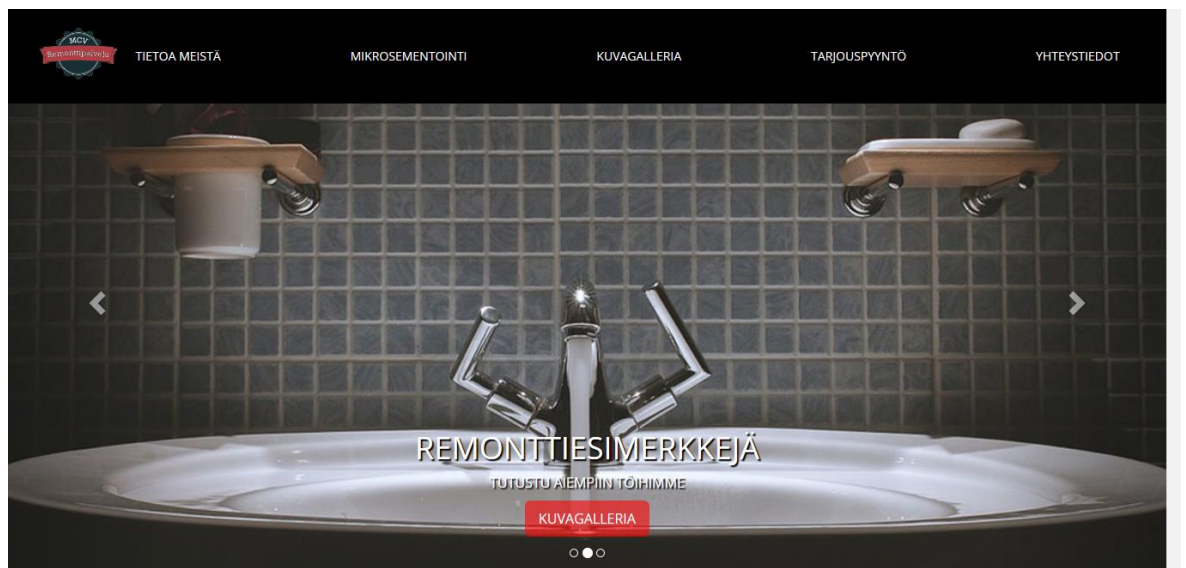
Empiirisen osan tuotos on julkaisuvalmis sivusto. Tämän projektin aikana sivustoa ei julkaistu, eikä sille suunniteltu mitään varsinaiseen julkaisemiseen liittyvää. Sivusto on yksisivuinen sivusto, joka rakentuu seitsemästä eri osiosta: navigointi, etusivu (kuvakaruselli), Tietoa meistä -osio, Mikrosementointi-osio, kuvagalleria, tarjouspyyntölomake sekä yhteystieto-osio/footer.

Navigointi sijaitsee sivuston ylälaudassa ja se on koko ajan näkyvillä. Navigointipalkissa olevaa yrityksen logoa klikkaamalla siirrytään aina sivuston ylälaitaan eli niin sanotulle etusivulle, jossa sijaitsee kuvakaruselli. Kuvakaruselli pyörittää kolmea eri kuvaa lyhyine teksteineen, joiden kautta pääsee siirtymään valittuihin osioihin sivustolla.

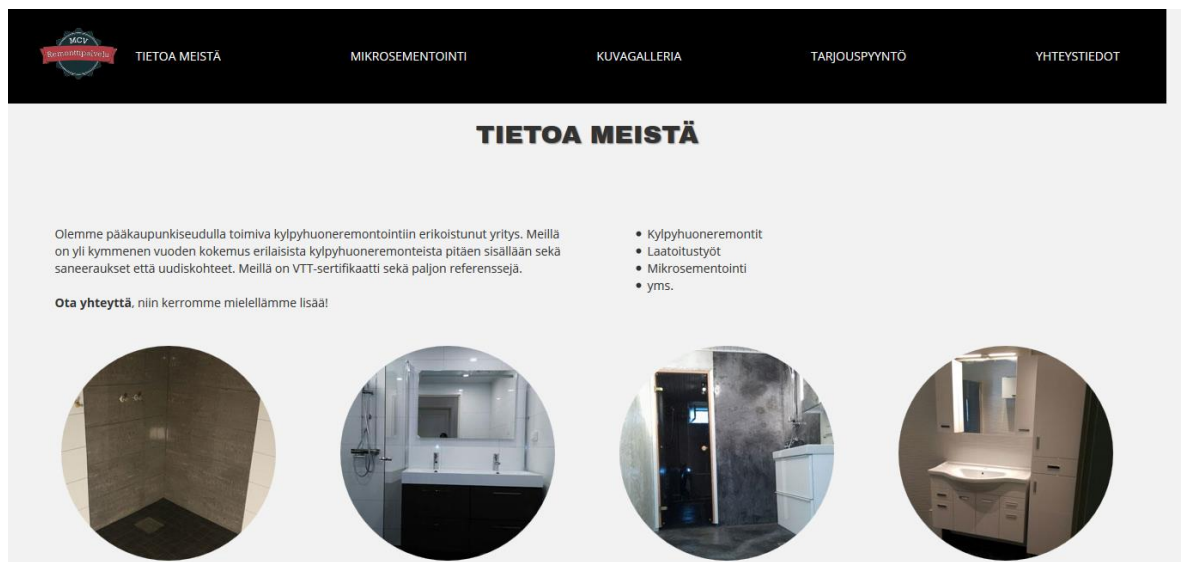
Sivustolla pääsee liikkumaan joko rullaamalla sivua alas- tai ylöspäin tai napauttamalla navigointipalkissa olevia linkkejä eri osioihin. Mikrosementointi-osiossa on linkki, jota napauttamalla selaimessa aukeaa uusi välilehti, jolla pääsee lukemaan tarkemmin yrityksen käyttämästä mikrosementointi-tuotteesta. Kuvagallerian kuvat saa avattua isommiksi napauttamalla kuvan auki ja halutessaan kuvia voi selata nuolinäppäinten avulla myös avatussa koossa.

Tarjouspyyntö-osiossa sijaitsee lomake, jota käyttämällä voi lähettää yritykselle haluamansa viestin. Lomakkeen vieressä on lisäohjeistusta tarjouspyyntöä tekevälle. Yhteystiedot-linkki vie sivuston alalaitaan eli footeriin, jossa lukee yrityksen yhteystiedot eli sähköpostiosoite ja puhelinnumero.

Alla olevissa kuvakaappauksissa näkyy lopullinen sivusto. Kuvat 57–63 kuvaavat sivuston Desktop-näkymän ja kuvat 64–71 sivuston responsiivisen näkymän. Responsiivisessa näkymässä näyttökoon pienentyessä navigointipalkki muuttuu alasvetovalikoksi ja osa teksteistä ja kuvista pienenee.



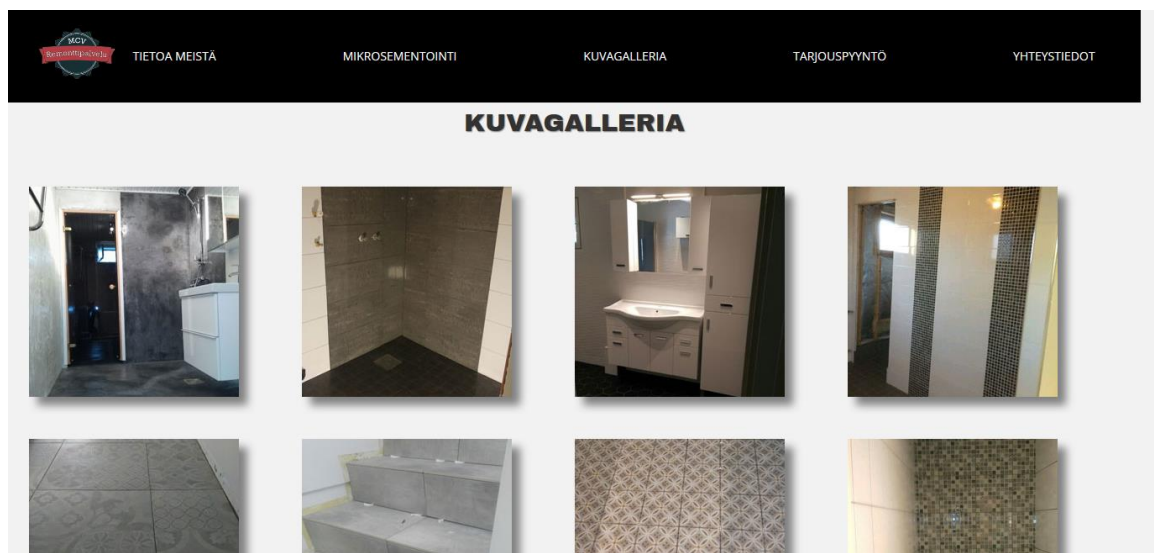
Kuva 57. Sivuston etusivu



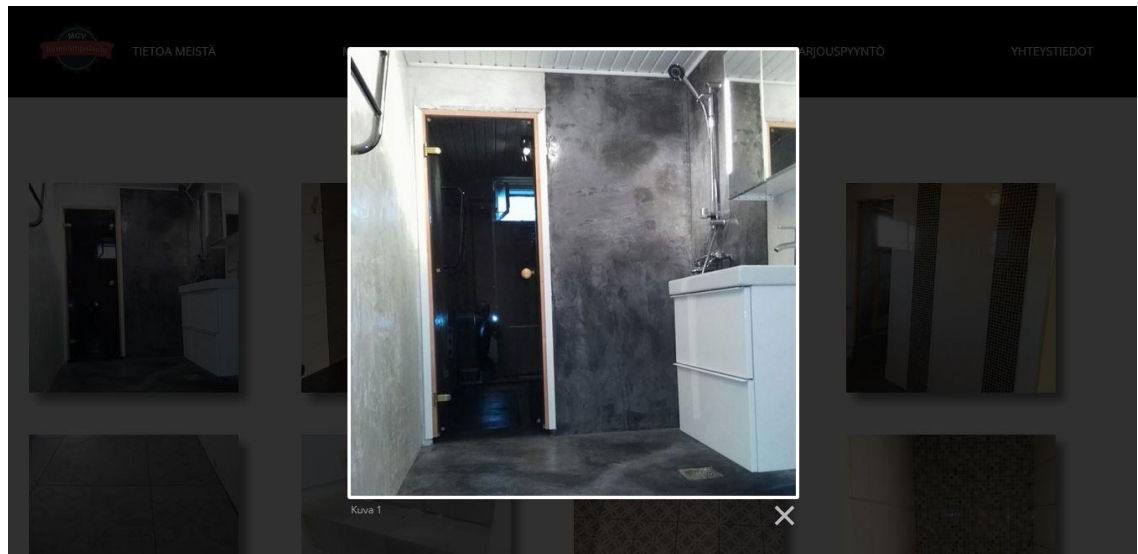
Kuva 58. Sivuston "Tietoa meistä" -sivu



Kuva 59. Sivuston "Mikrosementointi"-sivu



Kuva 60. Sivuston "Kuvagalleria"-sivu



Kuva 61. Sivuston kuvagallerian avattu kuva

WIC  
Käsitönteollisuus

Tietoa Meistä    Mikrosementointi    Kuvagalleria    Tarjouspyyntö    Yhteystiedot

## TARJOUSPYYNTÖ / YHTEYDENOTTO

NIMI:

OSOITE:

SÄHKÖPOSTI:

PUHELINNUMERO:

VIESTI:

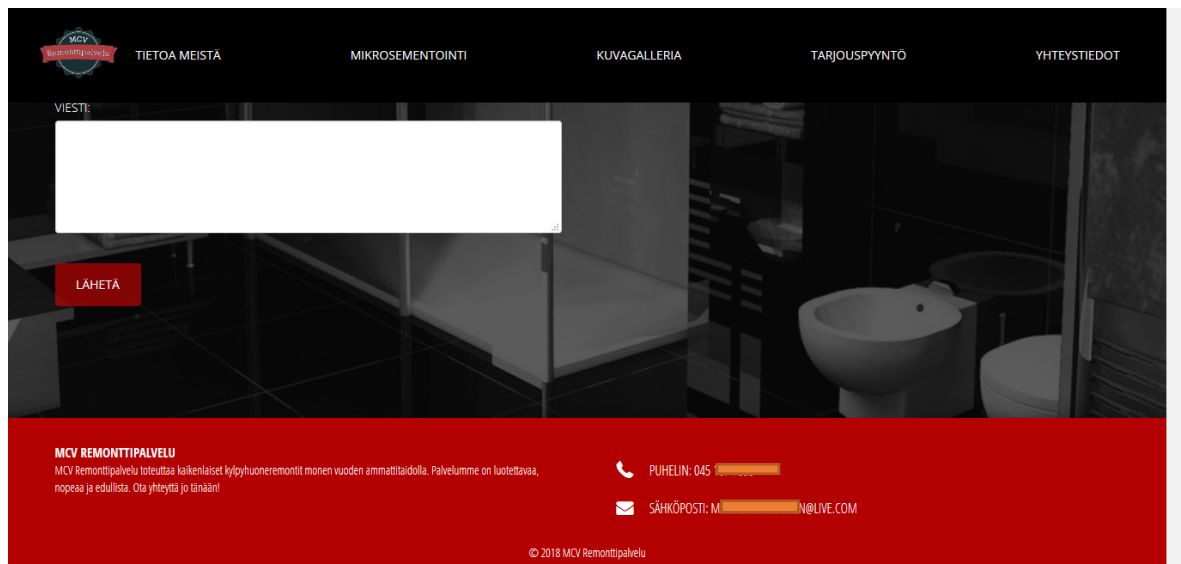
Annamme erittäin mielellämme lisätietoja yritykseemme sekä palveluihimme liittyen.

Mikäli haluat tehdä tarjouspyynnön, kuvaathan mahdollisimman tarkasti haluamasi työn. Ilmoita myös kohteen sijainti sekä kylpyhuoneen ja/tai remontoitavan alueen neliömäärä. Tarkennamme kohteen tietoja tarvittaessa lisäkysymyksillä ja näin pystymme antamaan hyvinkin tarkan tarjouksen. Tarvittaessa käymme tekemässä kohteen katselmuksen, mikäli emme muutoin saa riittävän tarkkaa kuvaa remontin laadusta.

Täytä oheinen lomake ja palaamme asiaan mahdollisimman pian!

Kuva 62. Sivuston "Tarjouspyyntö/yhteydenotto"-sivu





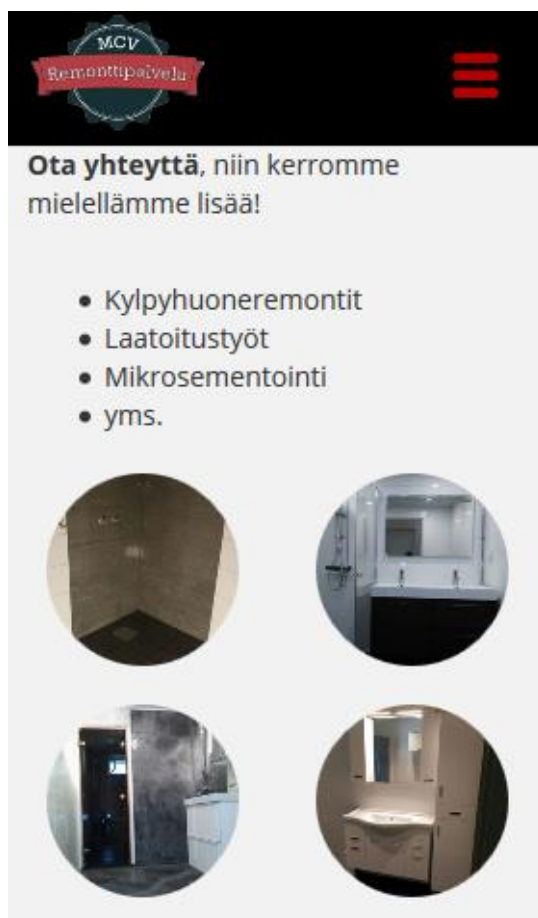
Kuva 63. Sivuston yhteystiedot/footer



Kuva 64. Sivuston etusivu mobiilinäytöllä



Kuva 65. Sivuston navigaatio mobiilinäytöllä



Kuva 66. Sivuston "Tietoa meistä" -sivu mobiilinäytöllä

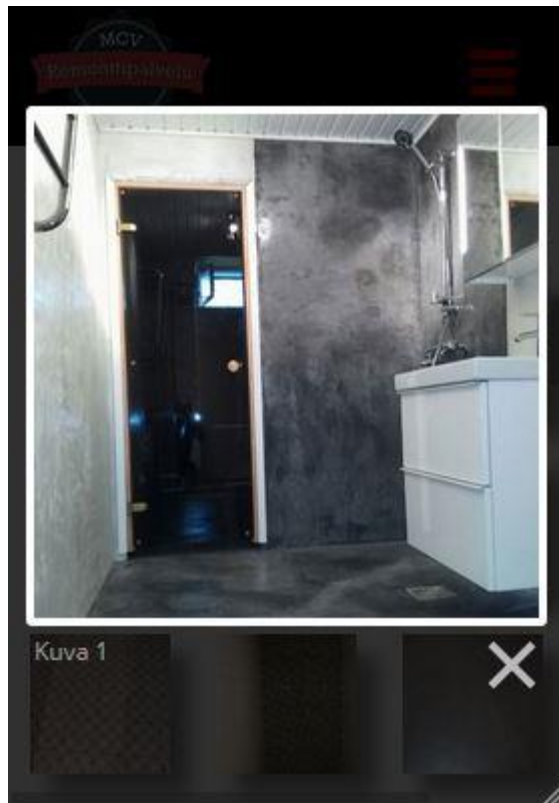


Kuva 67. Sivuston "Mikrosementointi"-sivu mobiilinäytöllä



Kuva 68. Sivuston "Kuvagalleria"-sivu mobiilinäytöllä





Kuva 69. Sivuston kuvagallerian avattu kuva mobiilinäytöllä

**TARJOUSPYYNTÖ /  
YHTEYDENOTTO**

NIMI:

OSOITE:

SÄHKÖPOSTI:

PUHELINNUMERO:

Kuva 70. Sivuston "Tarjouspyyntö/yhteydenotto"-sivu mobiilinäytöllä



Kuva 71. Sivuston yhteystiedot/footer mobiilinäytöllä

## 12 Pohdinta

Opinnäytetyön tuotoksena syntyi julkaisuvalmis verkkosivusto pienyritykselle. Tavoitteena oli suunnitella sivusto, joka toimisi mainoksena ja portfoliona toimeksiantajalle. Projekti eteni etukäteen tehdyn suunnitelman mukaisesti, mutta aikataulu kärsi samaan aikaan alkaneiden työharjoitteluiden ja muun elämän takia, joten projektin loppuun saattaminen viivästyi alkuperäisestä suunnitelmasta muutamalla kuukaudella. Asiakkaalle tämä ei kuitenkaan ollut ongelma ja projektin etenemisen hidastaminen tehtiin harkitusti ja yhteisymmärryksessä.

Työn alkaessa tehtiin vertailuanalyysiä eri remontointiyriytysten verkkosivujen välillä sekä pyrittiin myös suunnittelemaan sivuston käyttäjäkokemus hyväksi tekemällä think aloud -käytettävyydestä pienelle joukolle testaajia. Testi nostikin esiin paljon huomioita varhaisesta prototyypistä, joka oli rakennettu asiakkaan toiveiden mukaan. Asiakkaan toive oli kuitenkin, että verkkosivut toimisivat enemmänkin aiempien töiden esittelykanavana, portfoliona, kuin varsinaisena liiketoiminnan edistäjänä. Tämän takia monta testaajien toiveissa ollutta osiota jäi sivuilta pois ja sivusto jätettiin mahdollisimman yksinkertaiseksi, kuitenkin niin, että siinä oli kaikki ehdottoman tarpeellinen, kuten yhteystiedot ja kuvia aiemmista töistä. Mahdollisille uusille asiakkaille annettiin myös helppo tapa ottaa yhteyttä yritykseen lomakkeen avulla joko remontin tarjouspyyntöä tai muuta tiedustelua varten.

Tarve niinkin yksinkertaiselle sivustolle tuli hieman yllätyksenä, mikä aiheutti hieman päänvaivaa työn tekovaiheessa, mutta pyrkimys oli kuitenkin kunnioittaa asiakkaan toiveita samalla toteuttaen kuitenkin mahdollisimman toimiva ja tyylikäs sivusto.

Alusta lähtien tarkoitus oli, että asiakas tekee pieniä päivityksiä sivustoon itse, lähinnä kuvien päivitystä. Käyttöohjeiden tekeminen tuotti kuitenkin hieman ongelmia ja mahdollisia tapoja tehdä ohjeistus ja toteuttaa päivitykset käytiin läpi pitkään.

Projektia hankaloitti teorian suuri määrä ja sen hallitseminen, ettei teoriaperusta riistäytyisi täysin hallinnasta, vaan tarjoaisi tiiviin ja selkeän peruspaketin verkkosivujen kehitystyössä huomioon otettavista aihepiireistä. Hankalimpia teorian aiheita oli vaatimusmäärittely ja sen soveltaminen verkkosivujen kehittämiseen, sillä aiheesta löytyi ristiriitaista tietoa ja sitä oli hankala soveltaa verkkosivujen kehittämiseen.

Työmäärän jakaminen onnistui hyvin, vaikka koodin tekeminen oli ajoittain hankalaa erillään ja molempien työssä käymisen vuoksi näkeminen kasvokkain ei aina onnistunut. Kommunikaatio kuitenkin pysyi koko ajan hyvänä ja kummankin roolit kussakin vaiheessa

olivat molemmille selvät, eikä konfliktitilanteita syntynyt, vaan olimme melko hyvin samalla sivulla koko projektin ajan.

Jatkoa ajatellen sivustoa voi vielä kehittää eteenpäin. Tämän projektin ulkopuolelle jäi sivuston julkaiseminen ja kaikki siihen liittyvä, kuten sopivan webhotellin valinta ja julkaisuun liittyvät lisäykset koodiin. Jatkoa ajatellen myös tietoturva olisi hyvä ottaa huomioon, sitä ei tässä projektissa käsitelty lainkaan.

## Lähteet

Ansaharju, J. 2011. Segmentoi ja löydä markkinarakosi. Luettavissa: <http://www.sisalto-markkinointi.fi/2011/02/15/segmentoi-ja-loyda-markkinarakosi/>. Luettu: 26.2.2018.

Astra Design Company. History of Web Design. Luettavissa: [http://www.astra-design.com/web\\_design\\_history.html](http://www.astra-design.com/web_design_history.html). Luettu: 1.3.2018.

Bergström, S. & Leppänen, A. 2009. Yrityksen asiakasmarkkinointi. 13. uudistettu painos. Edita Publishing Oy. Helsinki.

Chapman, C. 2009. The Evolution of Web Design. Luettavissa: [https://web.archive.org/web/20131030030802/http://sixrevisions.com/web\\_design/the-evolution-of-web-design/](https://web.archive.org/web/20131030030802/http://sixrevisions.com/web_design/the-evolution-of-web-design/). Luettu: 1.3.2018.

Computer Hope 2017a. HTML. Luettavissa: <https://www.computerhope.com/jargon/h/html.htm>. Luettu: 20.2.2018.

Computer Hope 2017b. Håkon Wium Lie. Luettavissa: [https://www.computerhope.com/people/hakon\\_wium\\_lie.htm](https://www.computerhope.com/people/hakon_wium_lie.htm). Luettu: 20.2.2018.

Computer Hope 2017c. CSS. Luettavissa: <https://www.computerhope.com/jargon/c/css.htm>. Luettu: 20.2.2018.

Computer Hope 2017d. Brendan Eich. Luettavissa: [https://www.computerhope.com/people/brendan\\_eich.htm](https://www.computerhope.com/people/brendan_eich.htm). Luettu: 20.2.2018.

Computer Hope 2017e. JavaScript. Luettavissa: <https://www.computerhope.com/jargon/j/javascr.htm>. Luettu: 20.2.2018.

Derome, J. 2015. What is User Experience? Luettavissa: <https://www.usertesting.com/blog/2015/08/13/what-is-user-experience/>. Luettu: 21.2.2018.

Dhakar, L. Lightbox. Luettavissa: <http://lokeshdhakar.com/projects/lightbox2/>. Luettu: 20.2.2018.

Fontello. Fontello – icon fonts generator. Luettavissa: <http://fontello.com/>. Luettu: 8.1.2018.

Fulmore 2011. Mistä myyvät internetsivut on tehty – Osa 3: Kohderyhmä ja sisältö. Luettavissa: <https://fulmore.fi/uncategorized/mista-myyvat-internetsivut-on-tehty-osa-3/>. Luettu: 30.9.2017.

Gelbmann, M. 2018. Web Technologies of the Year 2017. Luettavissa: [https://w3techs.com/blog/entry/web\\_technologies\\_of\\_the\\_year\\_2017](https://w3techs.com/blog/entry/web_technologies_of_the_year_2017). Luettu: 15.2.2018.

Girard, J. 2017. Choosing the Right Content Management System (CMS). Luettavissa: <https://www.thoughtco.com/choosing-a-content-management-system-3466373>. Luettu: 1.3.2018.

Global Benchmarking Network 2018a. Benchmarking. Luettavissa: <http://www.globalbenchmarking.org/benchmarking/>. Luettu: 21.2.2018.

Global Benchmarking Network 2018b. Development of Benchmarking. Luettavissa: <http://www.globalbenchmarking.org/benchmarking/development-of-benchmarking/>. Luettu: 22.2.2018.

Global Benchmarking Network 2018c. Benefits of Benchmarking. Luettavissa: <http://www.globalbenchmarking.org/benchmarking/benefits-of-benchmarking/>. Luettu: 22.2.2018.

Global Benchmarking Network 2018d. Types of Benchmarking. Luettavissa: <http://www.globalbenchmarking.org/benchmarking/types-of-benchmarking/>. Luettu: 22.2.2018.

Global Benchmarking Network 2018e. Internal Benchmarking. Luettavissa: <http://www.globalbenchmarking.org/benchmarking/types-of-benchmarking/internal-benchmarking/>. Luettu: 22.2.2018.

Global Benchmarking Network 2018f. External Benchmarking. Luettavissa: <http://www.globalbenchmarking.org/benchmarking/types-of-benchmarking/external-benchmarking/>. Luettu: 22.2.2018.

Interaction Design Foundation. User Experience (UX) Design. Luettavissa: <https://www.interaction-design.org/literature/topics/ux-design>. Luettu: 21.2.2018.

ISO 2010. Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems. Luettavissa: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-210:ed-1:v1:en>. Luettu: 21.2.2018.

Itä-Suomen yliopisto. Benchmarking. Luettavissa: <https://www.uef.fi/benchmarking>. Luettu: 25.9.2017.

Ivanovs, A. 2017. WYSIWYG Website Builders for Online Business. Luettavissa: [http://www.huffingtonpost.com/alex-ivanovs/wysiwyg-website-builders\\_b\\_8814882.html](http://www.huffingtonpost.com/alex-ivanovs/wysiwyg-website-builders_b_8814882.html). Luettu: 1.3.2018.

Kaskela, L. 2005. Vaatimusmäärittely. Luettavissa: <https://www.tieke.fi/pages/viewpage.action?pageId=3441242>. Luettu: 2.4.2018.

Kauppinen, M. 2015. WordPress ja muut julkaisujärjestelmät vertailussa. Luettavissa: <https://www.meom.fi/wordpress-ja-muut-julkaisujarjestelmat-vertailussa/>. Luettu: 15.2.2018.

Kyrnin, J. 2017a. The Three Layers of Web Design. Luettavissa: <https://www.thoughtco.com/three-layers-of-web-design-3468761>. Luettu: 1.3.2018.

Kyrnin, J. 2017b. The Best Windows WYSIWYG HTML Editors. Luettavissa: <https://www.thoughtco.com/best-windows-wysiwyg-editors-3471324>. Luettu: 1.3.2018.

Kyrnin, J. 2018. 10 Best Free HTML Editors for Windows for 2018. Luettavissa: <https://www.thoughtco.com/best-free-html-editors-for-windows-3471313>. Luettu: 1.3.2018.

Kähönen, P. 2016. Perinteinen vaatimusmäärittely vs. käyttötapaukset. Luettavissa: <https://www.pasaati.com/blog/perinteinen-vaatimusm%C3%A4%C3%A4rittely-vs.-k%C3%A4ytt%C3%B6tapaukset>. Luettu: 2.4.2018.

Liljeroos, H. Internet-markkinoinnin ABC: Mitä on internet-markkinointi? Luettavissa: <http://www.digivallankumous.fi/internet-markkinoinnin-abc-mita-on-internet-markkinointi/>. Luettu: 29.9.2017.

Morville, P. 2004. User Experience Design. Luettavissa: [http://semanticstudios.com/user\\_experience\\_design/](http://semanticstudios.com/user_experience_design/). Luettu: 21.2.2018.

Norman, D. & Nielsen, J. 1998-2018. The Definition of User Experience (UX). Luettavissa: <https://www.nngroup.com/articles/definition-user-experience/>. Luettu: 21.2.2018.

Omni Partners Oy 2018. CMS eli Content Management System. Luettavissa: <https://omnipartners.fi/sanakirja/cms-eli-content-management-system/>. Luettu: 15.2.2018.

Ouellette, A. 2017. What is Bootstrap: A Beginners Guide. Luettavissa: <https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/>. Luettu: 20.2.2018.

Republica.fi. Kohdentaminen ja segmentointi. Luettavissa: <http://republica.fi/kohdentaminen-ja-segmentointi/>. Luettu: 26.2.2018.

Rouse, M. 2016. Content management system (CMS). Luettavissa: <http://searchcontentmanagement.techtarget.com/definition/content-management-system-CMS>. Luettu: 15.2.2018.

Ruluks, S. 2014. A brief history of web design for designers. Luettavissa: <http://blog.froont.com/brief-history-of-web-design-for-designers/>. Luettu: 1.3.2018.

Suomen Digimarkkinointi Oy 2018a. Mitä hakukoneoptimointi (SEO) on? Luettavissa: <https://www.digimarkkinointi.fi/blogi/mita-hakukoneoptimointi-seo-on>. Luettu: 27.2.2018.

Suomen Digimarkkinointi Oy 2018b. Verkkosivujen sisällön optimointi. Luettavissa: <https://www.digimarkkinointi.fi/blogi/verkkosivujen-sisallon-optimointi>. Luettu: 28.2.2018.

Tukes 2016. Turvallisuus- ja kemikaalivirasto. Tuotteiden käyttöohjeet ja turvallista käyttöä koskevat merkinnät. Opas. Mainostoimisto SST. Luettavissa: [http://www.tukes.fi/Tiedot/julkaisut/Tuotteiden\\_kaytto-ohjeet\\_opas.pdf](http://www.tukes.fi/Tiedot/julkaisut/Tuotteiden_kaytto-ohjeet_opas.pdf). Luettu: 28.2.2018.

Tuominen, E. 2013. Joomla, Drupal ja WordPress. Luettavissa: <https://genero.fi/joomla-drupal-ja-wordpress/>. Luettu: 15.2.2018.

Tutorial Republic 2018. Bootstrap Introduction. Luettavissa: <https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-introduction.php>. Luettu: 20.2.2018.

Tutorials Point 2018. What is CSS? Luettavissa: [https://www.tutorialspoint.com/css/what\\_is\\_css.htm](https://www.tutorialspoint.com/css/what_is_css.htm). Luettu: 23.4.2018.



VirtuaaliAMK a. Kirjalliset potilasohjeet viestinnän näkökulmasta. Luettavissa:  
<http://www2.amk.fi/digma.fi/eetu/www.amk.fi/opintojaksot/030905/1116425173436.html>.  
Luettu: 28.2.2018.

VirtuaaliAMK b. Mistä silmäiltävyys muodostuu? Luettavissa:  
<http://www2.amk.fi/digma.fi/eetu/www.amk.fi/opintojaksot/030905/1116425173436/1117079889682/1117096379185/1117097167387.html>.  
Luettu: 28.2.2018.

VirtuaaliAMK c. Millaista kieltä potilasohjeisiin? Luettavissa:  
<http://www2.amk.fi/digma.fi/eetu/www.amk.fi/opintojaksot/030905/1116425173436/1117079889682/1117096421323/1117097601734.html>.  
Luettu: 28.2.2018.

VirtuaaliAMK d. Milloin ohje on lukijan kannalta helppokäyttöinen? Luettavissa:  
<http://www2.amk.fi/digma.fi/eetu/www.amk.fi/opintojaksot/030905/1116425173436/1117079889682/1117096445579/1117097997493.html>.  
Luettu: 28.2.2018.

VirtuaaliAMK e. Miten ohjeeseen saa juonellisuutta? Luettavissa:  
<http://www2.amk.fi/digma.fi/eetu/www.amk.fi/opintojaksot/030905/1116425173436/1117079889682/1117096456923/1117098498046.html>.  
Luettu: 28.2.2018.

VirtuaaliAMK f. Laatimisprosessi. Luettavissa:  
<http://www2.amk.fi/digma.fi/eetu/www.amk.fi/opintojaksot/030905/1116425173436/1117079889682/1141365129925/1141365242596.html>.  
Luettu: 28.2.2018.

W3schools.com 1999-2018a. Luettavissa: <https://www.w3schools.com/>. Luettu: 28.2.2018.

W3schools.com 1999-2018b. W3Schools' Online Certification Program. Luettavissa:  
<https://www.w3schools.com/cert/default.asp>. Luettu: 28.2.2018.

W3schools.com 1999-2018c. HTML Introduction. Luettavissa:  
[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp). Luettu: 20.2.2018.

W3schools.com 1999-2018d. JavaScript Tutorial. Luettavissa:  
<https://www.w3schools.com/js/default.asp>. Luettu 23.4.2018.

W3schools.com 1999-2018e. PHP 5 Introduction. Luettavissa:  
[https://www.w3schools.com/php/php\\_intro.asp](https://www.w3schools.com/php/php_intro.asp). Luettu: 27.2.2018.

W3schools.com 1999-2018f. Bootstrap Carousel Plugin. Luettavissa:  
[https://www.w3schools.com/bootstrap/bootstrap\\_carousel.asp](https://www.w3schools.com/bootstrap/bootstrap_carousel.asp). Luettu: 16.12.2017.

W3Techs 2009-2018a. Usage of content management systems for websites. Luettavissa:  
[https://w3techs.com/technologies/overview/content\\_management/all/](https://w3techs.com/technologies/overview/content_management/all/). Luettu: 15.2.2018.

W3Techs 2009-2018b. Site Info - Wordpress.com. Luettavissa: <https://w3techs.com/sites/info/wordpress.com>. Luettu: 15.2.2018.

Wikipedia 2017. Lightbox (JavaScript). Luettavissa: [https://en.wikipedia.org/wiki/Lightbox\\_\(JavaScript\)](https://en.wikipedia.org/wiki/Lightbox_(JavaScript)). Luettu: 20.2.2018.

Yrityksen-perustaminen.net 2018. Verkkokaupan perustaminen ilmaiseksi tai halvalla. Luettavissa: <https://yrityksen-perustaminen.net/verkkokaupan-perustaminen-ilmaiseksi/>. Luettu: 29.9.2017.

## Liitteet

### Liite 1. Keskeiset käsitteet

<b>Benchmarking</b>	Vertailuanalyysi ja vertaiskehittäminen – menetelmä, jossa yrityksen toimintaa ja prosesseja vertaillaan muihin yrityksiin.
<b>Bootstrap</b>	Verkkosovelluskehityksen työkalu, joka sisältää HTML-, CSS- ja JavaScript-koodia, joilla voi muun muassa tehdä helposti responsiivisia sivustoja.
<b>CSS</b>	<b>C</b> ascading <b>S</b> tyle <b>S</b> heets – yksinkertainen mekanismi, jolla kuvataan verkkosivun tyyli, kuten värit, fontit ja asemoinnit.
<b>Domain</b>	Verkkosivuston nimi, esimerkiksi google.com.
<b>Flash (Adobe Flash)</b>	Adobe Systemsin kehitysympäristö, jolla voi luoda erilaista multimediaa, kuten kuvagallerioita, animaatioita ja pelejä esimerkiksi verkkosivustoille ja mobiililaitteisiin.
<b>HTML</b>	<b>H</b> yper <b>T</b> ext <b>M</b> arkup <b>L</b> anguage – verkkosivujen luomiseen käytettävä merkintäkielistandardi.
<b>JavaScript</b>	HTML:n ja Internetin ohjelmointikieli.
<b>jQuery</b>	JavaScript-kirjasto, joka yksinkertaistaa JavaScript-ohjelmointia.
<b>Käyttäjäkokemus</b>	User Experience, UX – ihmisen huomioidut ja reaktiot liittyen tuotteen, järjestelmän tai palvelun käyttöön tai odotettuun käyttöön.
<b>Lightbox</b>	JavaScript-kirjasto, joka avaa kuvat ja videot näytölle himmentäen muun sivun taustalle.
<b>Media query</b>	CSS-tekniikka, jolla lisätään tietoa, miten sivusto reagoi, kun esimerkiksi mennään tietyn sivuleveyden tai -pituuden alle tai käännetään laite vaak- tai pystyasentoon.
<b>MySQL</b>	Verkossa käytettävä avoimen lähdekoodin tietokantajärjestelmä.
<b>PHP</b>	<b>P</b> HP: <b>H</b> ypertext <b>P</b> reprocessor (alun perin <b>P</b> ersonal <b>H</b> ome <b>P</b> age) – palvelinpuolen ohjelmointikieli, jolla voi esim. luoda dynaamista sisältöä verkkosivuille ja kerätä lomakkeiden tietoja.
<b>Responsiivisuus</b>	Verkkosivujen mukautuminen käyttäjän näytön koon mukaan, jolloin sivustoa on miellyttävä selailla sekä pöytäkoneella että mobiililaitteilla, kuten älypuhelimella ja tabletilla.
<b>Segmentointi</b>	Kohderyhmän määrittely.
<b>SEO</b>	<b>S</b> earch <b>E</b> ngine <b>O</b> ptimization – hakukoneoptimointi, verkkosivujen sisällön parantaminen hakusanoilla, jotta hakukoneet löytäisivät sivuston paremmin.

<b>Sisällönhallinta-järjestelmä</b>	<b>Content Management System</b> eli <b>CMS</b> – verkkopohjainen sovellus, jolla luodaan ja hallitaan nettisivuston sisältöä, esimerkiksi WordPress, Joomla ja Drupal.
<b>Think aloud</b>	Käytettävyydesti, jossa käyttäjä testaa järjestelmän käytettävyyttä suorittamalla tehtäviä ja miettimällä ääneen ("think aloud") valvojan tehdessä muistiinpanoja.
<b>URL</b>	<b>Universal Resource Locator</b> – verkkosivuston täydellinen osoite, esimerkiksi <a href="http://www.google.com">http://www.google.com</a> .
<b>Vaatimusmäärittely</b>	Menetelmä, jolla määritetään, miksi projekti on tarpeellinen ja mitä tarpeita sillä tulee tyydyttää.
<b>Verkkoselain</b>	Ohjelma, jolla voi selata www-sivuja. Esimerkiksi Google Chrome, Internet Explorer ja Mozilla Firefox.
<b>WYSIWYG-editori</b>	<b>What You See Is What You Get</b> – ohjelmisto, jolla sivuston tekijä näkee, miltä sivusto tulee näyttämään samalla kun sivustoa vasta luodaan.
<b>XHTML</b>	<b>EXtensible HyperText Markup Language</b> – merkintäkieli, joka on lähes identtinen HTML:n kanssa, mutta sisältää tarkempia sääntöjä, määrittellen XML-sovellukseksi.
<b>XML</b>	<b>EXtensible Markup Language</b> – merkintäkieli, joka varastoi ja siirtää tietoa ja on sekä ihmisen että koneen luettavissa.

## Liite 2. Projektisuunnitelma

Toteutuksen päävaiheet	
vk 1-2	1. Asiakkaan tarpeiden ja kohderyhmän kartoitus <ul style="list-style-type: none"><li>- palaveri asiakkaan kanssa</li><li>- tarvittavien materiaalien esim. kuvien pyytäminen asiakkaalta ohjelmointivaihetta varten</li></ul>
vk 3-4	2. Muiden vastaavien yritysten verkkosivustojen analysointi ja sen hyödyntäminen projektissa <ul style="list-style-type: none"><li>- Tutkitaan Internetistä vastaavien yritysten verkkosivustoja ja niiden heikkouksia ja vahvuuksia. Huomioita hyödynnetään sivuston suunnittelussa ja toteuttamisessa.</li></ul>
vk 5-6	3. Prototyyppi <ul style="list-style-type: none"><li>- sivuston mahdollisimman tarkka hahmottelu paperilla ja PowerPointillä ja sen hyväksyttäminen asiakkaalla</li></ul>
vk 7	4. Think aloud -käytettävyydestä prototyypille <ul style="list-style-type: none"><li>- prototyypin testaaminen ulkopuolisilla</li></ul>
vk 8	5. Korjaukset prototyyppiin think aloud -testin mukaan <ul style="list-style-type: none"><li>- korjataan prototyyppiä käytettävyydestä esiin tulleiden kommenttien/heikkouksien pohjalta</li></ul>
vk 9-18	6. Sivuston ohjelmointi <ul style="list-style-type: none"><li>- ohjelmointi HTML-, CSS-, JavaScript- ja Bootstrap-tekniikoita hyödyntäen</li><li>- kuvien ja muun materiaalin kerääminen asiakkaalta ja niiden käsittely</li><li>- sivuston läpikäynti asiakkaan kanssa</li></ul>
vk 19	7. Think aloud -käytettävyydestä sivustolle <ul style="list-style-type: none"><li>- sivuston testaaminen ulkopuolisilla</li></ul>
vk 20	8. Tarvittavat korjaustoimenpiteet sivustolle <ul style="list-style-type: none"><li>- korjataan sivustoa käytettävyydestä esiin tulleiden kommenttien/heikkouksien pohjalta</li></ul>
vk 21	9. Sivuston luovuttaminen asiakkaalle <ul style="list-style-type: none"><li>- sivuston hyväksyttäminen ja luovutus asiakkaalle</li></ul>
Tuotos	Empiirisen osan tuotos on julkaisuvalmis sivusto.

### Liite 3. Käyttöohjeet: GIMP – Asennus

Tämä ohje on tehty Windows-käyttöjärjestelmälle sekä Firefox-selaimelle ja asennuskieleksi on englanti. Muita käyttöjärjestelmiä ja selaimia käytettäessä esimerkiksi asennustiedostojen tallentaminen ja suorittaminen eroaa hieman ohjeista.

1. Mene sivulle <https://www.gimp.org> ja valitse **Download**, kuten kuvassa 1. Mikäli etusivulla ei ole kuvassa näkyvää oranssia Download-painiketta, valitse ylävalikosta Download.



Kuva 1. Siirry eteenpäin valitsemalla Download

2. Valitse oma käyttöjärjestelmäsi kohdasta **Show downloads for** ja aloita ohjelman lataus kuvassa 2 näkyvästä oranssista **Download directly** -painikkeesta. Valitse avautuvasta ikkunasta **Tallenna tiedosto**.

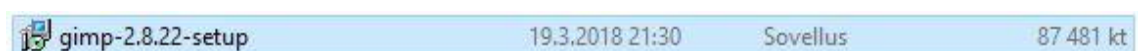
Show downloads for GNU/Linux | OS X | Microsoft Windows | All

#### GIMP for Windows



Kuva 2. Oranssi latauspainike

3. Hae koneelta ladatun ohjelman sijainti. Mikäli et tiedä mihin ohjelma latautui, etsi koneelta tiedostoa nimellä gimp. Kuvassa 3 näkyy ohjelman asennustiedoston nimi. Valitse kyseinen tiedosto kaksoisklikkaamalla, jolloin ohjelman asennus alkaa.



Kuva 3. Ohjelman asennustiedosto

4. Mikäli näytölle ilmestyy ikkuna ”**Sallitko tämän ohjelman tehdä muutoksia tähän laitteeseen?**”, vastaa **Kyllä**. Tämän jälkeen seuraa näytölle ilmestyviä ohjeita. Valitse haluamasi kieli → **Ok**. Valitse kuvan 4 mukaisesta avautuvasta ikkunasta **Install**. Ohjelman asennus kestää hetken ja sen valmistuttua valitse **Finish**, kuten kuvassa 5.



Kuva 4. Aloita asennus valitsemalla Install



Kuva 5. Viimeistele asennus valitsemalla Finish

5. Ohjelma on nyt asennettu koneellesi ja on valmis käytettäväksi. Jos et löydä ohjelman sijaintia, etsi taas nimellä gimp. Huomaa, että asennettu käyttövalmis ohjelma on nimellä gimp, joka on eri tiedosto kuin aiempi ohjelman asennustiedosto (gimp-2.8.22-setup). Ohjelma käynnistyy gimp-tiedostoa kaksoisklikkaamalla. Ohjelman todennäköisin sijainti on koneen C-levy ja sen alta löytyvä Program Files -kansio, kuten kuvassa

6 (Tämä tietokone → C: → Program Files → GIMP 2 → bin → gimp-2.8). Ohjelma käynnistyy kaksoisklikkaamalla kohdetta **gimp-2.8**. Huomaa, että numerot ohjelman perässä ovat versionumeroita ja voivat vaihdella ohjelman asennusajankohdasta riippuen.

Tämä tietokone > TI31205500A (C:) > Program Files > GIMP 2 > bin

Nimi	Muokauspäivä	Tyyppi	Koko
bzip2	6.5.2017 20:39	Sovellus	44 kt
dbus-1.dll	6.5.2017 20:46	Sovelluslaajennus	524 kt
freebl3.dll	6.5.2017 20:46	Sovelluslaajennus	330 kt
gimp-2.8	10.5.2017 18:40	Sovellus	5 368 kt

Kuva 6. Ohjelman sijainti koneella



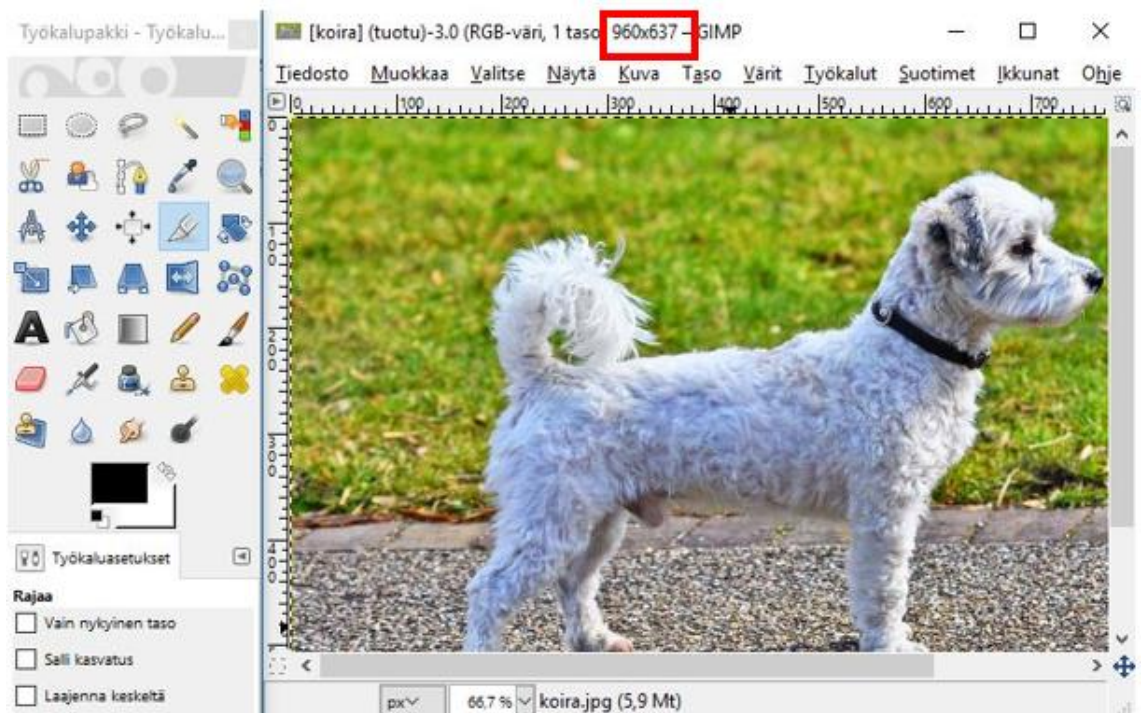
#### Liite 4. Käyttöohjeet: GIMP – Kuvankäsittely

Kuvankäsittelyohjelmaa tarvitaan lisättäessä uusia kuvia verkkosivujen kuvagalleriaan. Kuvagalleria koostuu kahdesta erikokoisesta kuvasta: pienestä, sivuilla koko ajan näkyvästä kuvasta, sekä suuresta kuvasta, joka avautuu pientä kuvaa klikkaamalla. Kuvat tulee muokata oikean kokoisiksi ennen niiden lisäystä kuvagalleriaan. Kuvagalleriaan tarvittavat kuvakoot ovat seuraavat:

- pieni: 250 x 250 px
- suuri 747 x 747 px.

#### Kuvan avaaminen GIMP-kuvankäsittelyohjelmalla

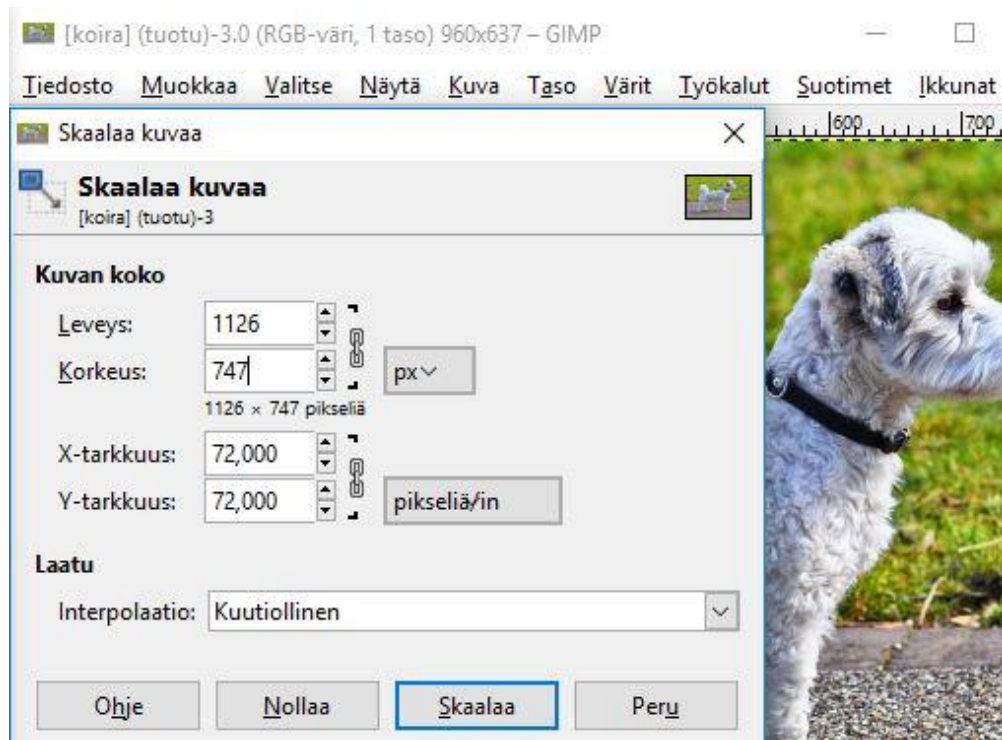
1. Tallenna muokattava kuva koneellesi.
2. Avaa GIMP. GIMP koostuu kolmesta toisistaan irrallisesta ikkunasta. Keskeisin ikkunoista sisältää avatun kuvan ja vasemmalla puolella sijaitsee työkalupakki, jota tulee myös tarvitsemaan. Voit liikutella ikkunoita ottamalla hiirellä kiinni ikkunan yläreunasta.
3. Avaa tallentamasi kuva valitsemalla **Tiedosto** → **Avaa**. Hae tallentamasi kuvan sijainti, valitse kuva → **Avaa**. Kuva avautuu ruudulle.
4. Kuvan koko on ilmoitettu kuvan yläpuolella, kuten kuvassa 1.



Kuva 1. Kuvan koko

## Kuvan muokkaus kokoon 747 x 747 px

1. Esimerkkinä käytettävän kuvan koko on 960 x 637 px. Kyseisen kuvan korkeus (637) ei siten ole riittävä kuvagalleriaan vaadittavaan suuren kuvan kokoon. Näin ollen kuva vaatii ensin muokkausta suuremmaksi. Huomaa, että kuvasi saattaa olla myös suurempi kuin 747 x 747, jolloin se vastaavasti saattaa vaatia muokkausta pienemmäksi tai mahdollisesti pelkkä rajausta riittää. Mikäli kuvasi ei vaadi koon muokkaamista, siirry suoraan kohtaan 4.
2. Valitse kuvan yläreunan valikkoriviltä **Kuva** → **Skaalaa kuvaa**. Kirjoita korkeuden kohdalle 747, kuten kuvassa 2, ja paina Enter. Kuvan leveys muuttuu automaattisesti oikeaksi suhteessa kuvan korkeuteen. Valitse **Skaalaa**. Kuvan koko muuttuu suuremmaksi, ja uudet mitat ovat jälleen nähtävillä kuvan yläpuolella.



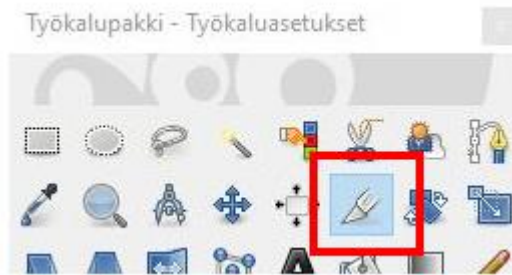
Kuva 2. Kuvan koon muuttaminen suuremmaksi

3. Kuvalla on nyt vaadittava korkeus, mutta sen leveys on liian suuri, kuten kuvasta 3 selviää. Seuraavaksi kuva tulee rajata oikeaan kokoon.



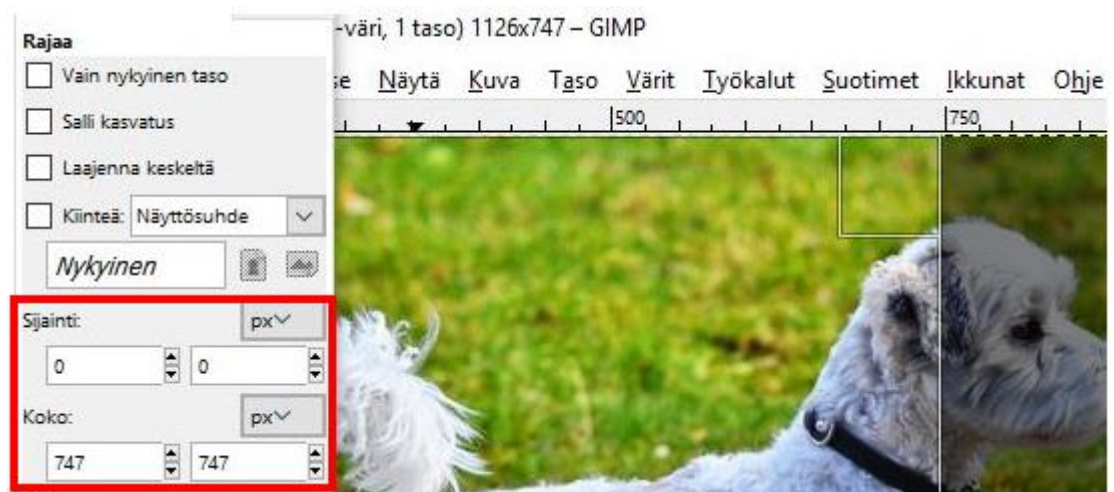
Kuva 3. Suurennetun kuvan koko

4. Rajataksesi kuvaa, valitse rajaustyökalu, kuten kuvassa 4. Löydät saman työkalun myös yläreunan valikkoriviltä reittiä **Työkalut** → **Kääntötyökalut** → **Rajaa**.



Kuva 4. Rajaustyökalu

5. Vie hiiren osoitin kuvan vasempaan ylänurkkaan, paina hiiren oikea painike alas ja liikuta hiirtä oikealle alaviistoon valitaksesi alueen kuvasta. Tässä vaiheessa valinnan koolla ei ole merkitystä, koska pääset muokkaamaan sitä seuraavassa vaiheessa. Vapauta hiiren painike.
6. Vasemmalla sijaitsevasta työkalupakki-ikkunasta näet sekä valinnan sijainnin että koon. Varmista, että molemmat sijainnin arvot ovat 0, jotta valinta lähtee täsmälleen kuvan vasemmasta ylänurkasta. Voit muuttaa arvot itse ja hyväksyä ne painamalla Enter. Aseta myös uudet arvot koko-kohtaan (747 x 747 px), jotta rajauksen koko vastaa haluttua. Paina arvojen muutosten jälkeen Enter, jotta uudet mitat aktivoituvat, kuten kuvassa 5.



Kuva 5. Rajaustyökalulla tehdyn valinnan sijainti ja koko

7. Voit vaihtaa kuvasta valittavaa aluetta viemällä hiiren valitun alueen päälle, painamalla hiiren oikea painike alas ja liikuttamalla valintaa hiirellä. Valinnan sijaintia on esimerkiksi tässä tilanteessa tarpeen muuttaa, jotta koiran pää ei jää kuvan ulkopuolelle. Kun

olet tyytyväinen valinnan sijaintiin, napauta kertaalleen hiiren oikealla painikkeella valinnan päällä. Kuva rajautuu määrittelemiisi mittoihin, kuten kuvassa 6.



Kuva 6. Rajatun kuvan uudet mitat

8. Tallentaaksesi kuvan valitse **Tiedosto** → **Vie nimellä**. Anna kuvallasi kuvaava nimi, kuten kuvassa 7, mutta älä koske kuvan loppupäätteeseen. Valitse tallennuskohde, joka on verkkosivujesi img-kansio → **Vie** → **Vie**.

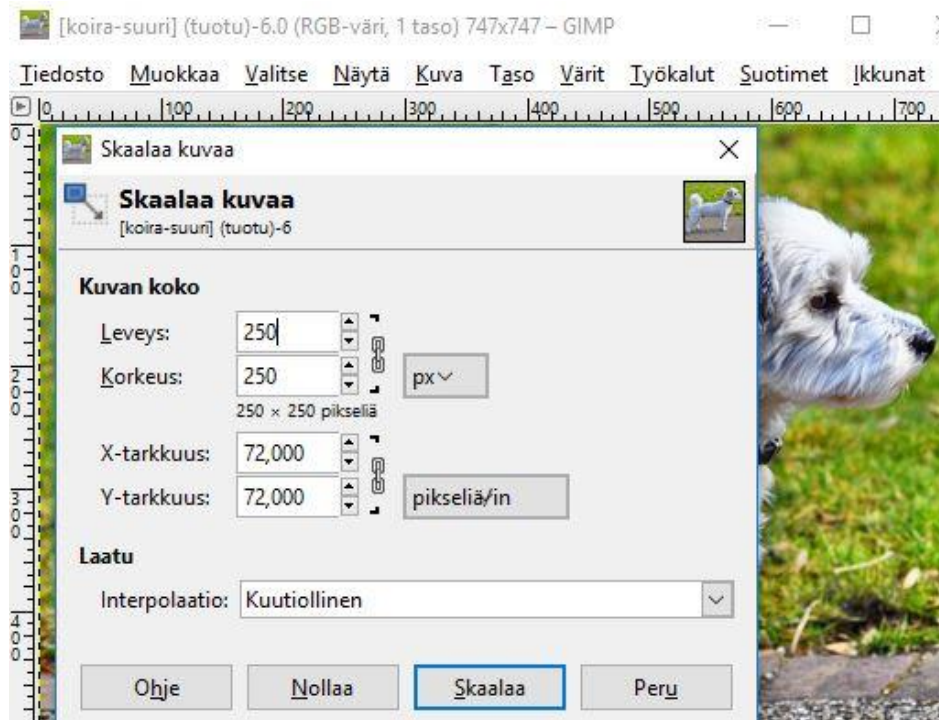


Kuva 7. Kuvan nimeäminen tallennusta varten

### Kuvan muokkaus kokoon 250 x 250 px

1. Avaa edellä tallentamasi koon 747 x 747 px kuva valitsemalla **Tiedosto** → **Avaa**. Hae tallentamasi kuvan sijainti, valitse kuva → **Avaa**. Kuva avautuu ruudulle.
2. Kuvan suhteet ovat valmiiksi oikeat, joten kuva tarvitsee ainoastaan muokkausta pienemmäksi. Valitse kuvan yläreunan valikkoriviltä **Kuva** → **Skaalaa kuvaa**. Kirjoita leveyden tai korkeuden kohdalle 250, kuten kuvassa 8, ja paina Enter. Toisen sivun mitta muuttuu automaattisesti samansuuruiseksi. Valitse **Skaalaa**. Kuva muuntuu pienemmäksi ja uudet mitat (250 x 250 px) ovat jälleen nähtävillä kuvan yläpuolella kuvan mukaisesti.





Kuva 8. Kuvan koon muuttaminen pienemmäksi



Kuva 9. Uuden kuvan mitat

3. Tallentaaksesi kuvan valitse **Tiedosto** → **Vie nimellä**. Anna kuvallasi kuvaava nimi, kuten kuvassa 10, mutta älä koske kuvan loppupääätteeseen. Valitse tallennuskohde, joka on verkkosivujesi img-kansio → **Vie** → **Vie**.



Kuva 10. Kuvan nimeäminen tallennusta varten

## Liite 5. Käyttöohjeet: Notepad++ – Asennus

Tämä ohje on tehty Firefox-selaimelle ja asennuskielenä on suomi. Muita selaimia käytettäessä esimerkiksi asennustiedostojen tallentaminen ja suorittaminen eroaa hieman ohjeista.

Jotta sivuille voi tehdä muutoksia, kuten lisätä kuvia, tarvitaan tekstieditori, jolla pääsee käsittelemään sivujen lähdekoodia. Notepad++ on ilmainen ja helppokäyttöinen tekstieditori ja soveltuu hyvin tähän tarkoitukseen.

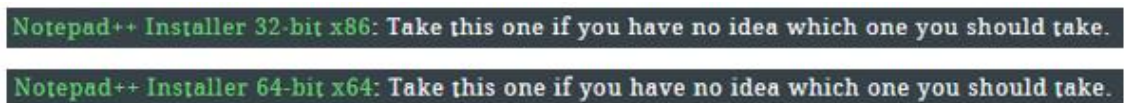
### Notepad++ -ohjelman asennus

1. Kirjoita Internet-selaimesi osoiteriville <https://notepad-plus-plus.org/>.
2. Valitse **download** ja sen alta avautuva **Current Version**, joka tarkoittaa ohjelman uusinta versiota (kuva 1).



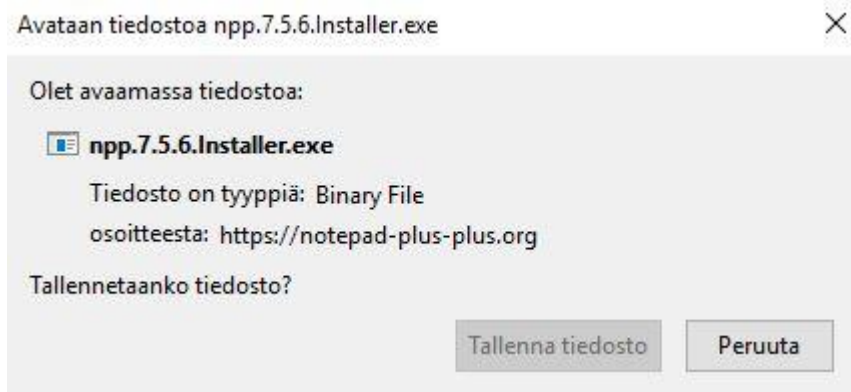
Kuva 1. Notepad++ -ohjelman lataus

3. Valitse listalta versio sen mukaan, onko käyttöjärjestelmäsi 32- vai 64-bittinen (kuva 2).



Kuva 2. Oikean asennustiedoston valinta

4. Valitse avautuvasta ikkunasta Tallenna tiedosto (kuva 3).



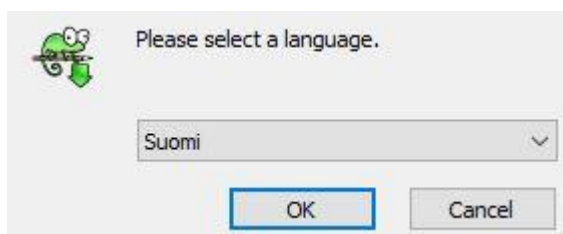
Kuva 3. Asennustiedoston tallennus

5. Etsi lataamasi tiedoston sijainti koneeltasi. Oletuksena on, että tiedosto tallentuu koneesi C-aseman ladattuihin tiedostoihin, jonne pääset navigoimaan reittiä Tämä tietokone → C: → Käyttäjät → [Käyttäjän nimi] → Ladatut tiedostot (kuva 4). Aloita ohjelman asennus kaksoisklikkaamalla npp.7.5.6.Installer-riviä. Huomaa, että tiedoston numerot voivat vaihdella asennusajankohdan mukaan, joten numeroiden ei tarvitse täsmätä tässä ohjeessa näkyviin.



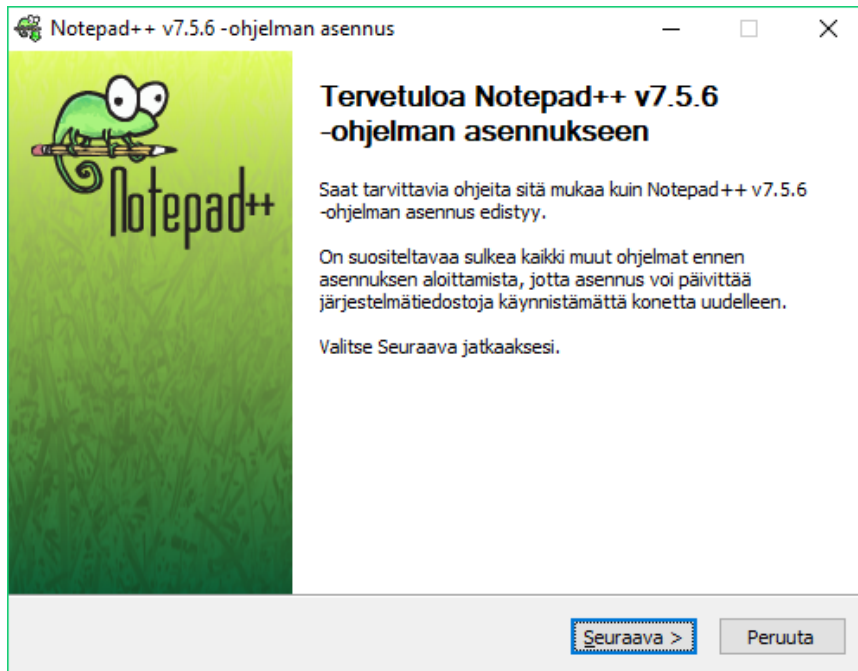
Kuva 4. Asennustiedoston avaus

6. Mikäli näytölle ilmestyy ikkuna ”**Sallitko tämän ohjelman tehdä muutoksia tähän laitteeseen?**”, vastaa **Kyllä**.
7. Valitse haluamasi kieli → **Ok** (kuva 5).



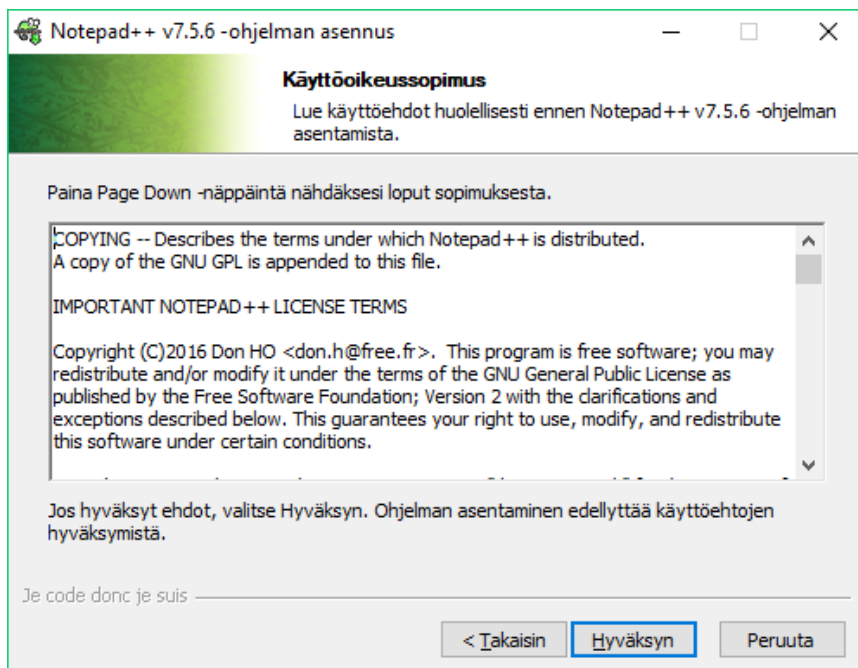
Kuva 5. Kielen valinta

8. Aloita ohjattu asennus valitsemalla **Seuraava** (kuva 6).



Kuva 6. Ohjelman asennuksen aloittaminen

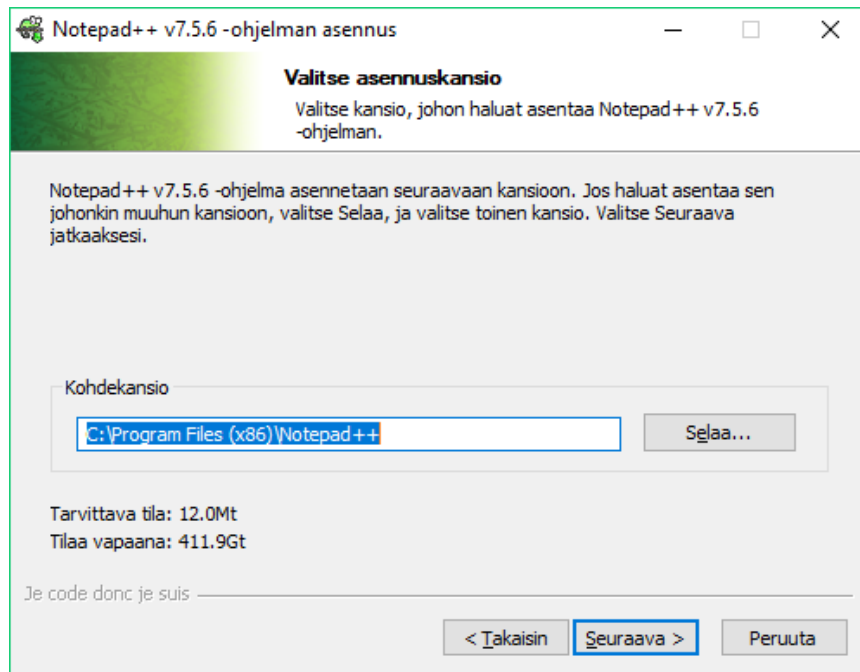
9. Seuraava ikkuna käsittelee käyttöoikeussopimusta. Lue käyttöehdot ja valitse **Hyväksyn** (kuva 7).



Kuva 7. Käyttöehtojen hyväksyminen

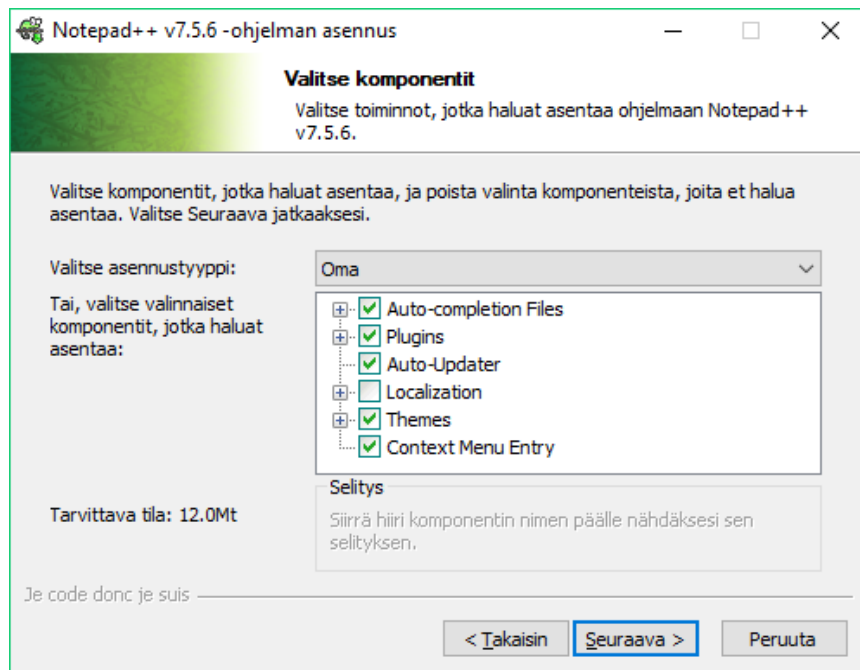


10. Valitse asennuskansio: Voit halutessasi muuttaa kohteen asennuskansion sijaintia, mutta se on tarpeetonta → **Seuraava** (kuva 8).



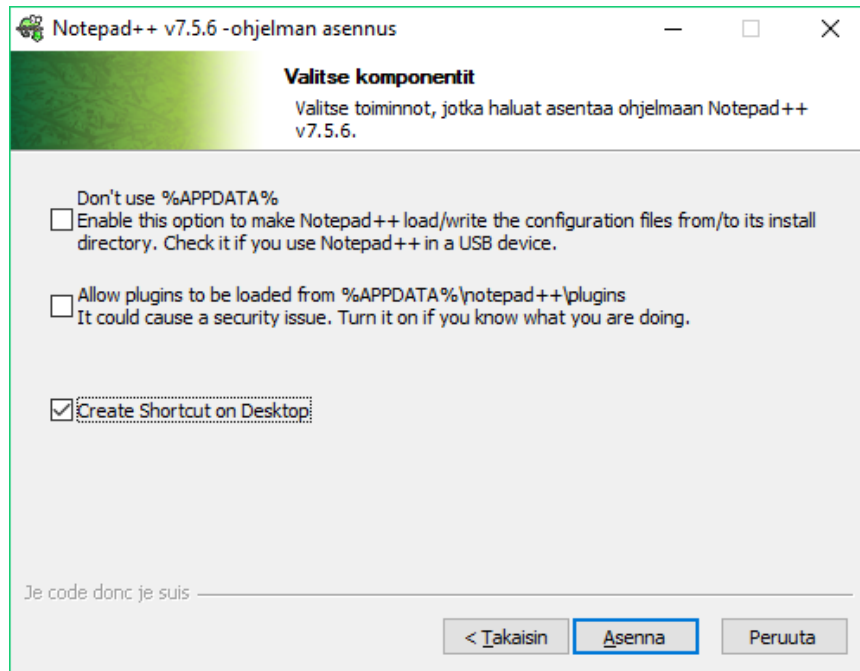
Kuva 8. Asennuskansion valinta

11. Valitse komponentit: Myöskään tähän kohtaan ei ole tarpeellista tehdä muutoksia → **Seuraava** (kuva 9).



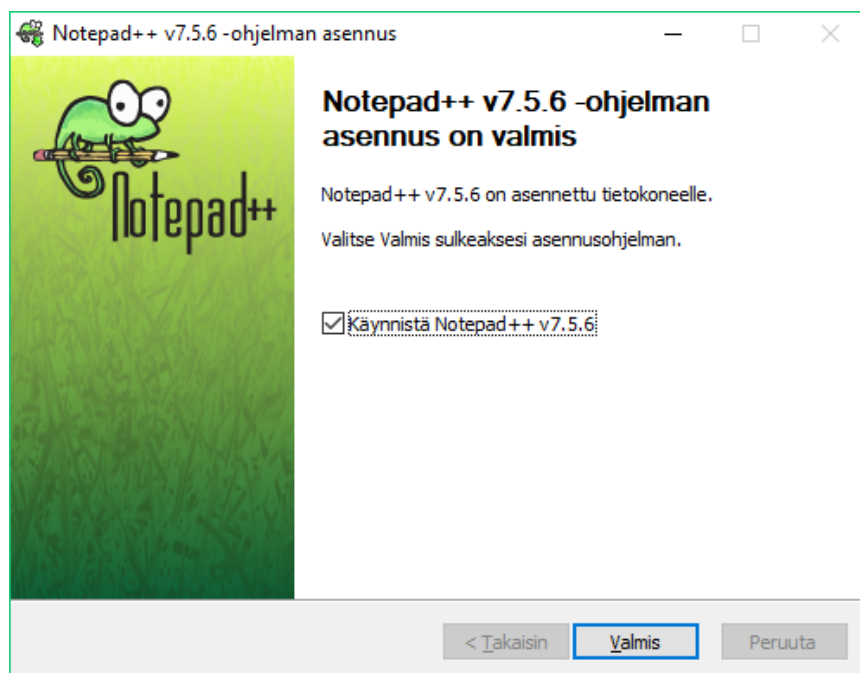
Kuva 9. Komponenttien valinta

12. Valitse komponentit: Valitse kohta **Create Shortcut on Desktop** → **Asenna** (kuva 10). Create Shortcut -valinta luo ohjelman pikakuvakkeen työpöydällesi, josta ohjelma on helpointa käynnistää.



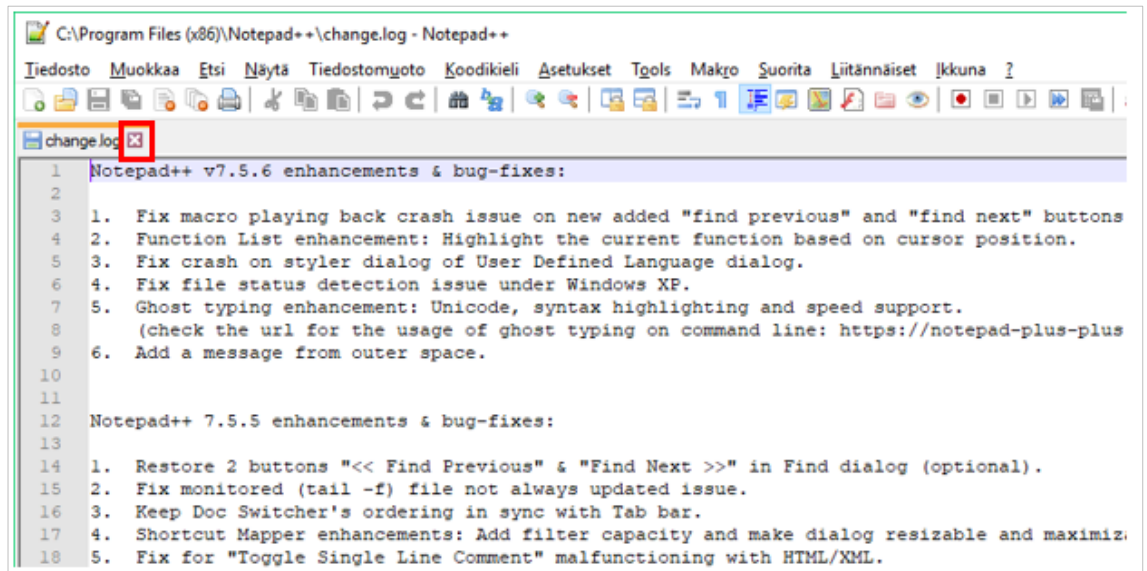
Kuva 10. Pikakuvakkeen luominen ja ohjelman asennus

13. Ohjelman asennus vie hetken ja sen valmistuttua saat siitä ilmoituksen (kuva 11). Valintaruutu **Käynnistä Notepad++** on valittuna, jolloin ohjelma käynnistyy automaattisesti valitessasi **Valmis**.

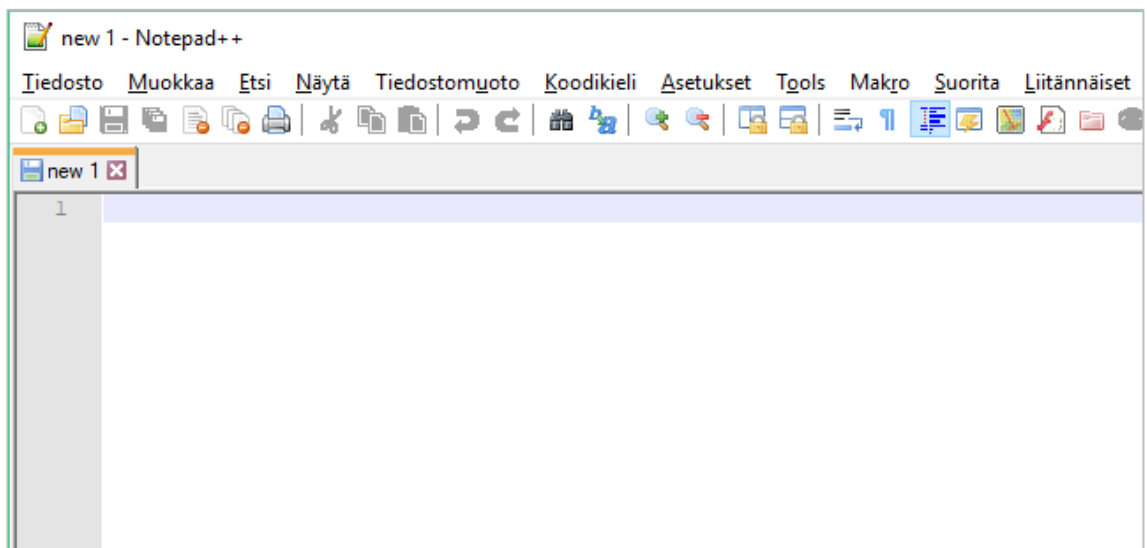


Kuva 11. Valmis asennus

14. Useimmiten ohjelman avaamisen yhteydessä avautuu pienempi ikkuna Notepad++ Plugin Manager. Jos näin käy, voit sulkea ikkunan valitsemalla Cancel.
15. Notepadin auetessa ohjelman ikkunaan tulee change.log-tiedosto (kuva 12), jossa kerrotaan asennetun version muutoksista ja korjauksista. Voit sulkea tiedoston painamalla punaisesta ruksista, jolloin näytölle jää uusi, tyhjä tiedosto new 1 (kuva 13).



Kuva 12. Change.log-tiedosto



Kuva 13. New 1 -tiedosto

## Liite 6. Käyttöohjeet: Notepad++ – Tiedoston käsittely

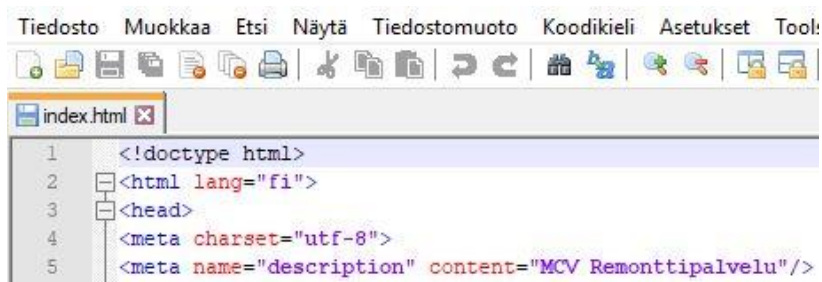
### Tiedoston avaaminen ja muutosten tallennus Notepad++ -ohjelmassa

1. Avaa Notepad++ kaksoisklikkaamalla työpöydälläsi olevaa kuvaketta (kuva 1).



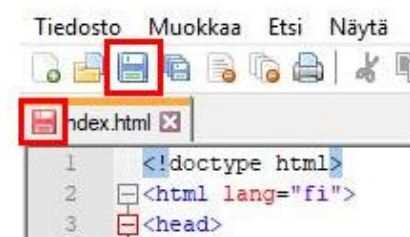
Kuva 1. Notepad++ -kuvake

2. Muokataksesi sivustosi koodia sinun tulee tehdä muutoksia sivuston index.html-tiedostoon. Hae index.html-tiedosto valitsemalla ohjelman yläriviltä **Tiedosto** → **Avaa**. Hae avautuvasta ikkunasta verkkosivujesi koodin sisältävän kansion sijainti. Oletetaan esimerkin vuoksi, että kansio on työpöydälläsi nimellä mcv. Valitse tällöin Työpöytä → mcv → kaksoisklikkaa tiedostoa index.html.
3. Näet index.html-tiedoston avattuna Notepad++ -ikkunassa (kuva 2).



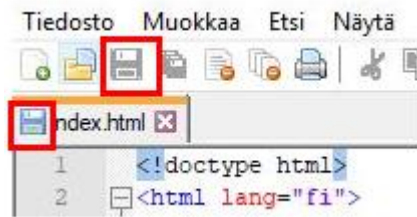
Kuva 2. Index.html-tiedosto

4. Sinun tulee tallentaa kaikki muutokset, jotka teet koodiin. Koodiin tehdyn muutoksen jälkeen tiedostonimen (index.html) edessä oleva kuvake muuttuu punaiseksi, mikä merkitsee sitä, ettei muutoksia ole tallennettu. Myös tämän yläpuolella oleva vastaava kuvake muuttuu aktiiviseksi. Tallenna muutokset aktiivisen sinisen kuvakkeen painalluksella (kuva 3).



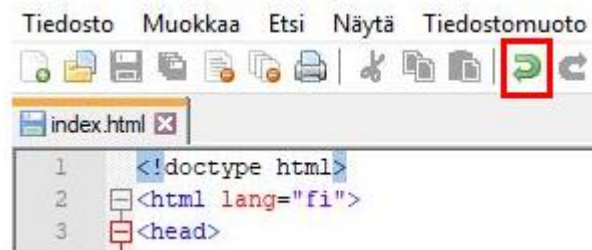
Kuva 3. Muutosten tallentaminen

5. Kun muutokset on tallennettu, muuttuu tiedostonimen edessä oleva kuvake siniseksi ja tämän yläpuolinen kuvake ei ole enää aktiivinen (kuva 4).



Kuva 4. Muutokset tallennettu

6. Mikäli haluat perua koodiin tekemäsi muutoksen, valitse Kumoa-painike (kuva 5). Näin tekemäsi muutokset palautuvat alkuperäiseen tilaan yksi kerrallaan. Kumoa-painike on turvallisempi vaihtoehto peruuttaa tehtyjä muutoksia, koska silloin ei vahingossa tule poistaneeksi mitään tärkeää merkistöä toimivan koodin kannalta.



Kuva 5. Muutosten kumoaminen

## Liite 7. Käyttöohjeet: Kuvagalleria

Kuvagalleria on alue, johon voi lisätä loputtomasti uusia kuvia. Uudet kuvat asettuvat vanhojen kuvien tapaan siten, että yhdelle riville mahtuu neljä kuvaa rinnakkain. Kuvagalleria koostuu pienistä sivuilla rinnakkain näkyvistä kuvista sekä suurista kuvista, jotka aukeavat hiiren valinnalla. Kuvagalleriaan lisättävän pienen kuvan koon tulee olla 250 x 250 px ja suuren kuvan 747 x 747 px. Kuvagalleriaan lisättävästä yhdestä uudesta kuvasta tulee siis olla kaksi erikokoista versiota: pieni ja suuri.

1. Muokkaa haluamasi kuva oikeaan kokoon (pieni ja suuri) ja tallenna molemmat yksilöllisillä nimillä verkkosivujesi img-kansioon. Tarkemmat ohjeet kuvankäsittelyyn löydät käyttöohjeiden kohdasta GIMP – Kuvankäsittely.
2. Avaa Notepad++ ja avaa verkkosivujesi index.html-tiedosto.
3. Etsi koodista kohta, jossa lukee KUVAGALLERIA ALKAA (kuva 1). Myös kuvagalleria-alueen loppuminen on merkitty koodiin samaan tyyliin (KUVAGALLERIA LOPPUU), joten tiedät, millä alueella koodia sinun tulee pysyä.

```
<!-- KUVAGALLERIA ALKAA -->  
<section class="galleria" id="kuva">  
  <div class="wrap">
```

Kuva 1. Merkintä kuvagalleria-alueen alkamisesta

4. Tarkastellessasi kuvagalleria-aluetta lähemmin huomaat, että siinä toistuu lähestulkoon samanlainen neljän rivin pätkä koodia kerta toisensa jälkeen (kuva 2). Jokainen tällainen neljän rivin pätkä koodia koskee yhtä kuvagallerian kuvaa, sisältäen sekä kuvan pienen että suuren version.

```
<div class="three column">  
  <a href="img/remontti-suuril.jpg" data-lightbox="gallery">  
    </a>  
</div>
```

Kuva 2. Ote gallerian koodista, joka sisältää yhden kuvan

5. Kopioi kuvan 2 mukainen pätkä koodia. Voit kopioida minkä tahansa vastaavan pätkän, mutta selkeyden vuoksi koodissa on merkintä kopioitavasta alueesta. Kyseinen merkintä ja kopioitava alue on nähtävillä kuvassa 3.

```
<!-- KOPIOI SEURAAVAT 4 RIVIÄ -->
<div class="three column">
  <a href="img/remontti-suurill.jpg" data-lightbox="gallery">
    </a>
  </div>
```

Kuva 3. Kuva kopioitavasta alueesta

6. Vie kursori kuvassa 3 näkyvän alimman rivin loppuun → Paina Enter → Kursori siirtyy seuraavalle tyhjälle riville.
7. Liitä kohdassa 5 kopioitu koodi muodostamallesi tyhjälle riville. Nyt sinulla on kaksi täysin samanlaista koodin pätkää allekkain (kuva 4). Kyseiset samanlaiset koodit peräkkäin tarkoittavat, että kuvagalleria-alueella esiintyy nyt sama kuva kaksi kertaa peräkkäin. Seuraavaksi sinun tulee muokata alempaa osaa koodista siten, että korvaat vanhat kuvat uusilla.

```
<!-- KOPIOI SEURAAVAT 4 RIVIÄ -->
<div class="three column">
  <a href="img/remontti-suurill.jpg" data-lightbox="gallery">
    </a>
  </div>
<div class="three column">
  <a href="img/remontti-suurill.jpg" data-lightbox="gallery">
    </a>
  </div>
```

Kuva 4. Koodi kopioituna ja liitettynä

8. Varmista tässä vaiheessa, että olet tallentanut haluamasi kuvat kahdessa eri koossa verkkosivujesi img-kansioon. Mikäli haluat käyttää samanlaista kuvien nimeämistapaa kuin kuvagallerian koodissa ennestään, voit nimetä kuvat esimerkiksi seuraavaan tyyliin:

- remontti-pieni12.jpg
- remontti-suuri12.jpg.

Voit vielä tässä vaiheessa muuttaa kuviesi nimeämistä, jos tahdot. Tärkeintä on nimetä kuvat siten, että nimet ovat yksilölliset, jolloin millään muulla kuvalla ei ole samaa nimeä. Saman kuvan pienen ja suuren version kannattaa myös olla helposti yhdistettävissä toisiinsa, kuten ylläolevassa nimeämisesimerkissä saman numeron avulla.

9. Korvaa kuvassa 5 merkittyjen kuvien nimet uusien kuvien vastaavilla nimillä. Ylemmälle riville tulee suuren kuvan nimi ja alemmalle vastaavasti pienen kuvan nimi. Muista olla koskematta mihinkään muuhun koodin merkistöön kuin kuvan nimeen. Uuden nimen tulee asettua tarkalleen ottaen kenoviivan (/) ja heittomerkin (") väliselle alueelle.

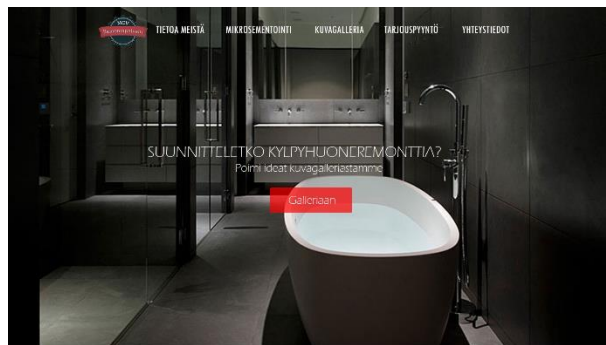
```
<!-- KOPIOI SEURAAVAT 4 RIVIÄ -->
<div class="three column">
  <a href="img/remontti-suurill.jpg" data-lightbox="gallery">
    </a>
  </div>
<div class="three column">
  <a href="img/remontti-suurill.jpg" data-lightbox="gallery">
    </a>
  </div>
```

Kuva 5. Alueet uusien kuvien nimien lisäämiselle

10. Tallenna tekemäsi muutokset. Tarkemmat ohjeet tallennukseen löytyvät käyttöohjeen kohdasta Notepad++ – Tiedoston käsittely.
11. Varmista muutoksen oikeellisuus avaamalla verkkosivusi selaimella. Huomaa, että sinun tulee päivittää sivut, mikäli ne ovat olleet muokkauksen ajan avoinna selaimessa.

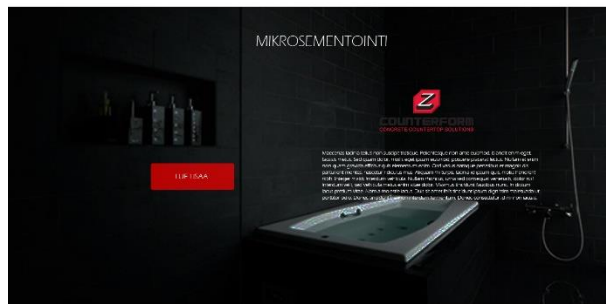


## Liite 8. Sivuston visuaalinen suunnitelma

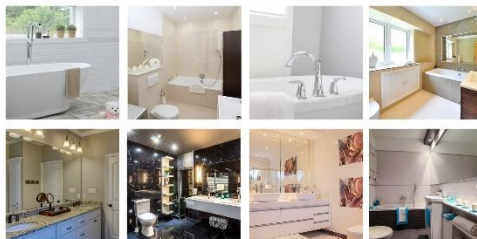


### TIEDOT MEISTÄ

Meidän työmme on se, että asiakas saa uuden, modernin kylpyhuonon. Pöytäidat ovat se, joka on asiakkaan mielestä tärkein osa remonttia. Pöytäidat ovat se, joka on asiakkaan mielestä tärkein osa remonttia. Pöytäidat ovat se, joka on asiakkaan mielestä tärkein osa remonttia.



### KUVAGALLERIA



LATAA LISÄÄ

